

NASA Technical Memorandum 104161

P-75

**COMPUTER PROGRAMS FOR THE CALCULATION OF
DUAL STING PITCH AND ROLL ANGLES
REQUIRED FOR AN ARTICULATED STING TO
OBTAIN ANGLES OF ATTACK AND
SIDESLIP ON WIND-TUNNEL MODELS**

John B. Peterson, Jr.

October 1991

(NASA-TM-104161) COMPUTER PROGRAMS FOR THE
CALCULATION OF DUAL STING PITCH AND ROLL
ANGLES REQUIRED FOR AN ARTICULATED STING TO
OBTAIN ANGLES OF ATTACK AND SIDESLIP ON
WIND-TUNNEL MODELS (NASA) 75 p CSCL 148 G3/09

N92-12007

Unclass
0052280



National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665-5225

INTRODUCTION

In many wind tunnels, the model support sting drive can both pitch and roll the sting support. In addition, some tunnels have articulated stings that are hinged between the sting drive mechanism and the model to pitch the model to higher angles. This allows tests to be made at higher angles of attack than those available with the regular sting drive.

Determination of the angles necessary to set the sting drive mechanism and the articulated sting pitch angle is simple when only a given angle of attack is desired. However, if an angle of sideslip is desired, it can be very difficult to determine the sting angles necessary to give the desired α and β on the model.

A program has been developed (ref. 1) to compute the pitch and roll of a conventional wind tunnel sting to position a model at the desired α and β . This program accepts stings with offset angles in yaw, pitch, and roll.

Also, an algorithm is described in reference 2 that uses matrix algebra to develop equations for the sting drive angles necessary to position a wind tunnel model in a wind tunnel. These equations can solve for three degree of freedom sting drive angles necessary to position the model at any attitude including the effects of sting offsets and sting bending angles.

The programs in this report compute the pitch and roll position of the conventional sting drive and the pitch of the high angle articulated sting to position the model at the desired α and β and position the model as near as

possible to the centerline of the tunnel.

Two computer programs have been developed and are described in this report. These programs cover the case of no accelerometers on board the model and the case of accelerometers on board the model to measure model pitch and/or model roll with respect to gravity.

These programs are iterative programs in that they calculate the α and β position of the model in the tunnel based on assumed pitch and roll angles of the sting drive and the pitch of the articulated sting; and then calculate new positions for the sting pitch and roll and second or articulated pitch to move the model closer to the desired α and β , and position the model in the center of the tunnel. This process is continued until the calculated α and β of the model match the desired α and β within a small increment and the model is close as possible to the center of the tunnel. This procedure insures that the α and β calculated by the data reduction program based on the available information such as stream flow angles, sting bending due to measured model forces, and on board measurements of model pitch and roll, if available, will match the desired α and β .

Both of these programs accept three sting offset angles, three sting bending angles, and two tunnel flow angles. In addition, the second program accepts on board measured pitch angle and on board measured roll angle, if available.

SYMBOLS

u	Velocity along X axis of model
v	Velocity along Y axis of model
w	Velocity along Z axis of model
X	Longitudinal body axis, positive aft, see figure 1
Y	Lateral body axis, positive to right, see figure 1
Z	Vertical body axis, positive upward, see figure 1
α	Model angle of attack, $\alpha = \arctan (w/u)$, see figure 1
β	Model angle of slideslip, $\beta = \arcsin (-v/V_\infty)$, see figure 1
θ	Model pitch angle, positive direction is nose up, see figure 1
θ_{off}	Sting offset angle in pitch
θ_{sb}	Sting bending in pitch
θ_{sc}	Sting drive pitch command
$\theta_{\text{sc}2}$	Articulated sting pitch command
ϕ	Model roll angle, positive direction is right wing down, see figure 1
ϕ_{off}	Sting offset angle in roll
ϕ_{sb}	Sting bending in roll
ϕ_{sc}	Sting drive roll command
$\phi_{\text{sc}2}$	Articulated sting roll
ψ	Model yaw angle, positive direction is nose right, see figure 1
ψ_{off}	Sting offset angle in yaw
ψ_{sb}	Sting bending in yaw

DETERMINATION OF ANGLE OF ATTACK AND ANGLE OF SIDESLIP FOR A WIND TUNNEL MODEL

The definition of angle of attack, α , on a wind-tunnel model is given in reference 3 as the arctan (w/u) where w is the component of the free stream velocity along the Z axis of the model (vertical axis with the positive direction upward) and u is the component of the free stream velocity along the X axis of the model (longitudinal axis with the positive direction aft). This definition applies no matter what the orientation of the model is. By examining the signs of u and w , the correct quadrant for α can be determined between -180° and $+180^\circ$. If both u and w are zero, then the angle of attack is indeterminate, but in these programs, when both u and w are equal to zero, α is defined to be equal to zero

The angle of sideslip, β , is defined in reference 3 as the arcsin ($-v/V_\infty$) where v is the component of the free stream velocity along the Y axis of the model (the lateral axis with the positive direction out the right or starboard wing) and V_∞ is the total free stream velocity. This means that β is positive when the flow is from the right.

In these computer programs, the free stream velocity, V_∞ , is set equal to one, and the components of the free stream velocity along the wind-tunnel axis system (u , v , and w) are calculated from the angles of upwash and sidewash in the wind tunnel. The upwash angle (UWA) is defined to be positive when the flow is upward in the tunnel (i.e., the w component is positive) and the sidewash to be positive when the flow is from the right to the left (from the starboard to the port). For positive wind-tunnel sidewash angle and the model pitch, roll and yaw angles equal to zero, the v component of the

model velocity is negative and the model sideslip angle, β , is positive.

The three velocities (u, v, and w) along with the three axes of the model are recalculated for each rotation of the model. By convention, the rotations are taken in order of yaw, pitch, and then roll (i.e., rotation about the Z axis, and Y axis, and then the X axis of the model). The equations for the u, v, and w velocities after rotation through each of the angles are given below.

Yaw (rotation about Z axis), ψ

$$U_A = U * \cos(\psi) - V * \sin(\psi)$$

$$V_A = V * \cos(\psi) + U * \sin(\psi)$$

$$W_A = W$$

Pitch (rotation about Y axis), θ

$$U_B = U_A * \cos(\theta) - W_A * \sin(\theta)$$

$$V_B = V_A$$

$$W_B = W_A * \cos(\theta) + U_A * \sin(\theta)$$

Roll (rotation about X axis), ϕ

$$U_C = U_B$$

$$V_C = V_B * \cos(\phi) - W_B * \sin(\phi)$$

$$W_C = W_B * \cos(\phi) + V_B * \sin(\phi)$$

Where ψ is the angle of yaw (positive for nose right), θ is the angle of pitch (positive for nose up), and ϕ is the angle of roll (positive for right wing down). These angles are shown in figure 1 from reference 4.

The equations given above are used in subroutine ALPBET of program

STNG2PR to calculate the velocities u , v , and w after each rotation of the model. After the three velocities are determined, the angle of attack, α , the angle of sideslip, β , are calculated using the formulas given above.

DESCRIPTION OF COMPUTER PROGRAMS

Two computer programs were developed to calculate the sting pitch angle, θ_{sc} , sting roll angle, ϕ_{sc} , and the second sting pitch angle, θ_{sc2} , necessary to obtain the desired α and β and to position the model as close as possible to the center line of the tunnel. Since this problem has three dependent variables (θ_{sc} , ϕ_{sc} , and θ_{sc2}), it requires that three independent quantities be specified in order that the problem be completely defined. The variables chosen are the α and β of the model and the position of the model in the tunnel with respect to the centerline of the tunnel. When no restraints are given for the three dependent variables (θ_{sc} , ϕ_{sc} , and θ_{sc2}), the program will find a solution that gives the desired α and β and puts the model on the centerline of the tunnel. However, this solution may require sting pitch angles, θ_{sc} , greater than or less than those obtainable from the tunnel sting pitch drive (see figure 2). In these cases, the program puts the sting at the limit of the sting drive and changes the other two variables (ϕ_{sc} and θ_{sc2}) to position the model at the desired α and β and as close as possible to the tunnel centerline.

The first program, STNG2, calculates the sting pitch and roll and second sting pitch angles based on the sting geometry, such as sting offsets and sting bending. This program consists of the main program and three subroutines. A listing of the program is given in Appendix A.

The second program, STNG2PR, calculates the sting pitch and roll and

second sting pitch angles from the sting geometry but also uses inputs from accelerometers on board the model that measure the model pitch and roll angles. This program consists of a main program and five subroutines and a listing is given in Appendix B.

A list and description of the variables used in these programs is given after the description of the programs.

PROGRAM STNG2

Program STNG2 calculates the sting pitch, θ_{sc} , sting roll, ϕ_{sc} , and second sting pitch, θ_{sc2} , necessary to obtain a desired model angle of attack, α , angle of sideslip, β , and position the model in the center of the tunnel. The program actually first calculates the α , β , and position of the model in the tunnel for an assumed sting pitch, sting roll, and second sting pitch. Then, by iteration, the assumed sting pitch, sting roll, and second sting pitch angles are changed until the calculated α and β agree with the desired or command α and β (ALPC and BETC) and the model is located on the centerline of the tunnel.

Program STNG2 is very similar to program STNG reported in reference 1. Program STNG2 has three factors, called FA, FB, and FD, which are equal to the change in alpha, beta, and distance from the centerline of the tunnel, respectively, with a unit change in the angle of sting (either pitch, θ_{sc} , roll ϕ_s , or second pitch angle, θ_{sc2} , depending on which angle is being optimized.) These factors are combined linearly with the error between the actual value of the angle or distance (ALP, BET, and DIS) and the desired angle or distance (ALPC, BETC, and zero distance) to give the function F. This

function is then minimized by subroutine CONV which uses a procedure which is equivalent to the secant method of finding the roots of an equation to find the sting position that reduces F to a minimum value.

Each time subroutine CONV is called to find the sting position to minimize F, subroutine VEL is called to determine the α and β , and subroutine DIS is called to determine the distance of the model from the centerline, at the new sting position. These new values for α , β , and distance are then used to calculate a new value for F. This process is repeated until the absolute value of F is less than the tolerance for F (TOLF = 0.00001) or for a maximum of three iterations.

The function F is calculated and minimized first for sting pitch (THESC), then for sting roll (PHISC), and then for the second sting pitch (THESC2).

After the F has been minimized for each of the three sting angles, the entire procedure is repeated until the change in each of the three sting angles (THESC, PHISC, and THESC2) is less than the tolerance for sting angles (TOLANG = 0.01) in the last iteration or for a maximum of 100 iterations.

The program usually reaches the limit for TOLANG in about 50 iterations with the α and β within about 0.001 degrees of the desired angle and the distance equal to about 0.2 (where the units for distance are the same as used to specify the sting lengths, R1, X2, and X3 at the beginning of the program).

The program then checks to see if the sting roll position is outside the

limits set for the sting roll position at the beginning of the program (-85° to 100° for model upright).

If the sting roll is outside the limits, the program is rerun from statement 50 with the pitch equal to negative of the pitch solution and the roll equal roll solution $\pm 180^\circ$. The solution with the new initial values for pitch and roll is very quick and the sting roll will be between the limits of -85° to 100°. If an inverted solution is desired, the limits on the sting roll can be changed to +80° for the lower limit (PHISLL) and 265° for the upper limit (PHISLU).

The program then checks to see if the sting pitch (THESC) is within the limits for sting pitch available in the wind tunnel. The lower limit, THESLL, is set to -11° and the upper limit, THESLU, is set to +19° in this program. If the sting pitch command THESC is within the sting limits, the program continues on to the section of the program starting at the DO 2000 statement where the angle of attack, ALP, and the angle of sideslip, BET, are converged to the commanded angle of attack, ALPC, and the commanded angle of sideslip, BETC, without regard to the distance of the wind tunnel model from the centerline of the tunnel. In this section of the program, the factor F is a function of the error in angle of attack, DALP and angle of sideslip, DBET only. Therefore,

$$F = FA * DALP + FB * DBET$$

As before, the function F is calculated and minimized first for the sting pitch (THESC), then for the sting roll (PHISC), and then for the second sting pitch (THESC2). This procedure is repeated until the error in angle of attack, DALP, and the error in angle of sideslip, DBET is reduced to less than the tolerance TOLDAB. (Set to 0.00001 at the beginning of the program).

If the sting pitch command, THESC, is outside the limits for sting pitch available in the wind tunnel, the sting pitch is set to the nearest limit and the program goes to statement 2020. In this part of the program as in the part after the DO 2000 statement, the program converges the angle of attack, ALP, and angle of sideslip, BET, to the commanded angles of attack and sideslip without regard to the distance from the wind tunnel model to the centerline of the tunnel. However, the sting pitch is fixed at the sting pitch limit and only the second sting pitch, THESC2, and the sting roll, PHISC, are used (in that order) to reduce the error in ALP and BET.

There is a counter in the program (IREPT) which is used to stop execution if the program gets into an infinite loop when it returns to statement 50 over and over. This can happen if the program is asked to solve an impossible case such as the BETC of 20° when the sting pitch limits are +19° and -11°.

Since STNG2 converges much more slowly than the STNG program did (sometimes 65 iterations in STNG2 compared to generally 10 to 20 in STNG), the answers from the previous calculation are saved and used as a starting point for the next calculations in the STNG2 program.

The answers for the first part of the program where THESC is unlimited are saved as the variables THESCUL, PHISCUL, and THES2UL. The answers of the last part of the program where THESC is limited are saved as the variable THESC2L and PHISCL.

The following is a list and description of the variables used in program STNG2 (all angles are degrees):

ALP	Angle of attack of model, α
ALPC	Command or desired angle of attack
BET	Angle of sideslip of model, β
BETC	Command or desired angle of sideslip
DALP	Differences between angle of attack of the model and desired angle of attack
DBET	Difference between angle of sideslip of the model and desired angle of sideslip
DIS	Distance of the wind tunnel model from the centerline of the wind tunnel measured perpendicular to the wing span (with $\phi_{sc2} = 0^\circ$).
DY	Lateral distance of the wind tunnel model from the centerline (Positive to the right)
DZ	Vertical distance of the wind tunnel model from the centerline (Positive upward)
F	Function which is to be minimized by pitching and rolling the sting drive and pitching the high angle articulated sting
FA	Weighting factor for DALP in the function F
FB	Weighting factor for DBET in the function F
FD	Weighting factor for DIS in the function F
IREPT	Variable used to limit the number of times the program starts over after converging outside the sting roll limits
PHIOFF	Sting offset angle in roll (ϕ_{off})
PHISB	Sting bending in roll ϕ_{sb})
PHISC	Sting roll command (ϕ_{sc}) required to position the model at the

	desired α and β and at the centerline of the tunnel
PHISCL	Sting roll command with sting pitch fixed at the limit. Used as a starting point for the next computation after statement 2020
PHISCUL	Sting roll command with sting pitch unlimited. Used as a starting point for the next computation just above statement 50
PHISC2	Second (articulated) sting offset in roll. Set to a constant value at the beginning of the program and not changed by the program.
PHISLL	Lower limit on sting roll.
PHISLU	Upper limit on sting roll.
PSIOFF	Sting offset angle in yaw (ψ_{off}).
PSISB	Sting bending in yaw (ψ_{sb}).
R1	Radius of the sting pitch arc sector (see figure 2).
SWA	Wind tunnel free stream sidewash angle, positive for flow from right.
THEOFF	Sting offset angle in pitch (θ_{off}).
THESB	Sting bending angle in pitch (θ_{sb}).
THESC	Sting pitch command (θ_{sc}) required to position the model at the desired α and β .
THESCUL	Sting pitch command with sting pitch unlimited. Used as a starting point for the next computation just above statement 50.
THESC2	Second (articulated) sting pitch angle.
THESC2L	Second sting pitch angle with sting pitch fixed at the limit. Used as a starting point for the next computation after statement 2020.
THESLL	Lower limit for sting pitch angle.
THESLU	Upper limit for sting pitch angle.
THES2UL	Second sting pitch angle with sting pitch unlimited. Used as a starting point for the next computation just above statement 50.
TOLANG	Convergence tolerance for the movement of the three sting angles

	during the last iteration.
TOLDAB	Convergence tolerance for the sum of the absolute values of DALP and DBET.
TOLF	Convergence tolerance for the value of F.
UWA	Wind tunnel free stream upwash angle, positive for flow from below.
u	Free stream velocity component in the longitudinal direction in the wind tunnel (the total free stream velocity is assumed to be 1.0).
v	Free stream velocity component in the lateral direction (flow from the right when looking forward is positive).
w	Free stream velocity component in the vertical direction (upward flow is positive).
X2	Distance from sting pitch arc sector to second (articulated) sting pitch axis. (See figure 2.)
X3	Distance from second sting pitch axis to wind tunnel model. (See figure 2.)

SUBROUTINE VEL

The purpose of subroutine VEL is to calculate the angle of attack, α , and angle of sideslip, β , of the wind tunnel model. The subroutine requires inputs of the velocity components in the wind tunnel u , v , and w ; the sting offset angles; the sting bending angles; and the sting drive angles. The subroutine calculates the components of the free stream velocity along the three axes of the model after each rotation angle using the formulas given in the section "Determination of Angle of Attack and Angle of Sideslip for a Wind Tunnel Model." After the last rotation, these velocities are used to calculate the angle of attack, α , and angle of sideslip, β , using the following formulas:

$$\alpha = \arctan (w/u)$$

$$\beta = \arcsin (-v/V_{\infty})$$

V_{∞} is set equal to one in the main program (STNG2) and therefore:

$$\beta = \arcsin (-v)$$

The following is a list and description of the additional variables used in subroutine VEL:

- UB, ... ,UI Longitudinal velocity in the model axis system after each rotation.
- VB, ... ,VI Lateral velocity in the model axis system after each rotation.
- WB, ... ,WI Vertical velocity in the model axis system after each rotation.

SUBROUTINE DISTANCE

Subroutine distance calculates the position of the wind tunnel model relative to the centerline of the wind tunnel. The subroutine requires inputs of the lengths of the various portions of the sting, R1, X2, and X3; the sting offset angles in yaw (PSIOFF), pitch (THEOFF) and roll (PSIOFF); and the sting drive angles, THESC, PHISC, and THESC2. The distance DY and DZ are calculated by subroutine DIST where DY is the lateral distance of the model from the wind tunnel centerline with positive direction to the right and DZ is the vertical distance from the centerline with positive values above the centerline. The units of the distances DY and DZ are the same as the units used to input the values of R1, X2, and X3.

The program starts with an axis system aligned to the model axis system and then translates and rotates the axis system along the stings until it is aligned to the wind tunnel axis system. As each of these translations and rotations are made, the subroutine calculates the position of the model in the axis system after each movement. Since the translations and rotations are taken in reverse order (normally the program calculates the effect of sting pitch first, and second sting roll last, (i.e., see subroutine VEL). The angles used in subroutine DIST are the negative of the angles used in the normal procedure.

The following is a list and description of the additional variables used in subroutine DIST:

- XA, ..., XJ Longitudinal distance of the wind tunnel model in the translated and/or rotated axis system (positive downstream).
- YA, ..., YJ Lateral distance of the wind tunnel model in the translated and/or rotated axis system (positive to the right facing upstream).
- ZA, ..., ZJ Vertical distance of the wind tunnel model in the translated and/or rotated axis system (positive upward).

SUBROUTINE CONV

The purpose of subroutine CONV is to minimize a function $Y = F(X)$. The function is calculated in the calling program and the subroutine calculates a new value of the independent variable X that will make the value of the dependent variable Y, nearer to zero. The new value of X is determined by calculating where a straight line through the last two previous pairs of points; X and Y, and XSAVE and YSAVE; intersects the X axis.

This new value is returned to the calling program in the X parameter and the old X and Y are placed in the XSAVE and YSAVE parameters at the end of the subroutine. In order to improve the stability of this procedure, certain limits are placed on the distance the new X value can be from the old X and XSAVE values. The IF statement above statement 3 of the subroutine CONV limits the new X value to be no more than 3.5 times the distance between the old X and XSAVE values away from the average of these values. Also if the two previous values for X are the same, the new X is calculated in a special way in statement 3. If the two previous Y values are the same, the new X is calculated to be the average of the two previous X's in statement 4.

The following is a list and description of the variables used in subroutine CONV:

DX	Absolute distance between X1 and X2.
X	Latest value of the independent variable. Also returned as the next value to be tried for the independent variable.
XA	Average of X1 and X2.
XK	Slope of line between X1, Y1 and X2, Y2.
XSAVE	Previous value of independent variable. Also returned as previous value of X.
X1	Previous value of the independent variable (same as XSAVE).
X2	Latest value of the independent variable.
Y	Latest value of the dependent variable.
YSAVE	Previous value of the dependent variable. Also returned as the previous value of Y.
Y1	Previous value of the dependent variable (same as YSAVE).
Y2	Latest value of the dependent variable.

PROGRAM STNG2PR

Program STNG2PR calculates the sting pitch and roll angles and articulated sting pitch angle required to obtain the desired angle of attack and angle of sideslip and position the model near the centerline of the wind tunnel when the model has on board measurements of the model pitch and roll (or pitch only if the roll is not available). The model pitch and roll is assumed to be measured relative to gravity.

At first it would appear that these cases would simplify the calculation of the alpha and beta of the model in the wind tunnel since the effects of sting offsets and sting bending are already included in the on board measurements of model pitch and roll. This is true in the case where a straight sting is used and the model yaw is assumed to equal zero. Since most wind tunnel tests use straight stings, on board measurements of model pitch and roll is a very valuable method of determining model alpha and beta in the wind tunnel. For bent stings, however, model yaw cannot be assumed to equal zero and model pitch and roll alone is not sufficient to determine model angle of attack and angle of sideslip.

In order to make program STNG2PR more generally applicable, no assumptions were made about the sting offset angles or the model yaw angle in the wind tunnel. Therefore, the program is applicable not only to the case of straight sting, but also to the case where the sting is offset and/or where sting bending occurs.

Program STNG2PR is similar to program STNG2 in that they both calculate the model α , β , and position of the model in the tunnel at a given

sting position and then, by the use of a factor, try to reduce the difference between the calculated α and β and the desired or commanded α and β and position the model near the centerline of the wind tunnel.

Program STNG2PR uses the same function (F) as program STNG2 uses to reduce the error between the calculated α and β and the desired α and β and to position the model near the centerline of the wind tunnel. The major difference between the two programs is that STNG2PR uses a three-step process to calculate α and β instead of the one step that STNG uses (subroutine VEL). The three steps are: first, calculate the yaw, pitch and roll angles of the model from the sting support system angles and the sting geometry (subroutine SIMUST), second, correct the pitch and roll of the model by the difference between the measured model pitch and roll, and the calculated model pitch and roll (DTHEMOB and DPHIMOB) determined at the beginning of the program; and third, calculate the α and β of the model from the model yaw, pitch and roll angles (subroutine ALPBET).

When programs STNG2 and STNG2PR are used in actual wind tunnel situations, the sting drive angles calculated by these programs will change somewhat as the sting and model move to the commanded positions because the sting bending angles will change as the angles of attack and sideslip of the model change, and because the differences between the calculated model pitch and roll and the on board measured model pitch and roll change as the model attitude changes. The final position, however, will be the correct position to obtain the desired α and β , since the final values for the sting bending and measured model pitch and roll will be the same as those used by the wind tunnel data reduction program.

The following is a list and description of the additional variables used in program STNG2PR (all angles are in degrees):

DPHIMOB	Difference between PHIMOB and PHIMT determined at the beginning of the program.
DTHEMOB	Difference between THEMOB and THEMT determined at the beginning of the program.
PHIMOB	Roll and model as measured by on board accelerometers.
PHIMOBT	Theoretical on board roll of model. $PHIMOBT = PHIMT + DPHIMOB$
PHIMT	Theoretical roll of model as determined by subroutine SIMUST from the sting drive angles, sting offset angles, and sting bending angles.
PHIS	Actual sting drive roll angle.
PHIS2	Actual articulated sting pitch angle.
PSIMT	Theoretical yaw angle of the model as determined by subroutine SIMUST from the sting drive angles, sting offset angles, and sting bending angles.
THEMOB	Pitch of model as measured by on board accelerometers.
THEMOBT	Theoretical on board pitch of model. $THEMOBT = THEMT + DTHEMOB$
THEMT	Theoretical pitch of model as determined by subroutine SIMUST from the sting drive angles, sting offset angles and sting bending angles.
THES	Actual sting drive pitch angle.
THES2	Actual articulated sting pitch angle.

SUBROUTINE SIMUST

Subroutine SIMUST simulates the sting-support, sting, and model system mathematically to calculate the model yaw, pitch and roll angles from the sting drive, sting offset and sting bending angles. The method of calculating the model yaw, pitch and roll is discussed in reference 4 and 5, and is explained below.

The pitch angle of the model is determined by calculating the X component, in the model axis system, of a unit vector in the Z direction of the tunnel axis system (XZ). The pitch angle is then the arcsin (-XZ). The pitch angle can range from -90° to $+90^\circ$.

The roll of the model is determined by calculating the Y and Z components, in the model axis system, of a unit vector in the Z direction of the tunnel axis system (YZ and ZZ). The roll of the model is then the arctan $(-YZ/ZZ)$ where the quadrant of the roll angle is determined by the signs of YZ and ZZ individually. The roll angle can range from -180° to 180° . If both YZ and ZZ are zero (i.e., the pitch angle is $\pm 90^\circ$) then the roll of the model is determined by the arctan $(-YX, ZX)$ and the yaw of the model is defined to be equal to zero (where YX and ZX are the Y component and the Z component respectively of a unit vector in the X direction of the tunnel axis system).

The yaw of the model is determined by calculating the X component in the model axis system of a unit Y vector in the wind tunnel axis system, XY, and the X component in the model axis system of a unit X vector in the tunnel axis system, XX. The yaw of the model is then the arctan $(-XY/XX)$ and can

range from -180° to 180° .

The following is a list and description of the additional variables used in subroutine SIMUST:

PHIS	Sting drive roll angle.
PHIS2	Second sting roll angle.
THES	Sting drive pitch angle.
THES2	Second sting drive pitch angle.
XX, YX, ZX	The X, Y, and Z components in the model axis system of a unit vector in the X direction in the tunnel axis system.
XY, YY, ZY	The X, Y, and Z components in the model axis system of a unit vector in the Y direction in the tunnel axis system.
XZ, YZ, ZZ	The X, Y, and Z components in the model axis system of a unit vector in the Z direction in the tunnel axis system.

SUBROUTINE COMP

Subroutine COMP is used to calculate the components, in the model axis system, of a unit vector in the wind tunnel axis system so that the model yaw, pitch and roll angles can be determined. This subroutine is very similar to subroutine VEL in program STNG2 which was used to calculate the velocity components in the model body axis system. Although this subroutine can calculate the X, Y, and Z components, in the model axis system, of an arbitrary vector in the wind tunnel axis system, it is only used in this program to calculate the components of a unit vector in the X, Y, or Z direction in the wind tunnel axis system. This means that one of the components, X, Y, or Z, is set equal to one and the other components are set equal to zero in the calling

program argument list. The components in the model body axis system of the unit vectors are then used in subroutine SIMUST to calculate the model yaw, pitch, and roll.

The following is a list and description of the additional variables used in subroutine COMP:

X, Y, Z	X, Y, and Z components of a vector in the tunnel axis system.
XB,...,XI	X component in the model axis system of a vector in the tunnel axis system after each rotation.
YB,...,YI	Y component in the model axis system of a vector in the tunnel axis system after each rotation.
ZB,...,ZI	Z component in the model axis system of a vector in tunnel axis system after each rotation.

SUBROUTINE ALPBET

The purpose of subroutine ALPBET is to calculate the angle of attack, α , and angle of sideslip, β , of the wind tunnel model. The subroutine requires inputs of the velocity components in the wind tunnel, U, V, and W, and the model Euler angles, yaw (ψ), pitch (θ), and roll (ϕ). The subroutine calculates the components of the free stream velocity along the three axes of the model after each rotation angle using the formulas given in the section "Determination of Angle of Attack and Angle of Sideslip for a Wind Tunnel Model." After rotation through the three Euler angles, the velocities are used to calculate the angle of attack, α , and angle of sideslip, β , using the following formulas:

$$\alpha = \arctan (v/u)$$

$$\beta = \arcsin (-v/V_{\infty})$$

V_{∞} is set to one in the main program (STNG2PR) and therefore:

$$\beta = \arcsin (-v)$$

The following is a list and description of the additional variables used in subroutine ALPBET (all angles are in degrees):

ALP	Angle of attack of model, α .
BET	Angle of sideslip of model, β .
PHIM	Angle of roll of model, ϕ .
PSIM	Angle of yaw of model, ψ .
THEM	Angle of pitch of model, θ .
UA, UB, UC	Longitudinal velocity component in the model axis system after each Euler angle rotation.
VA, VB, VC	Lateral velocity component in the model axis system after each Euler angle rotation.
WA, WB, WC	Vertical velocity component in the model axis system after each Euler angle rotation.

CONCLUDING REMARKS

Two programs have been developed to calculate the pitch, roll, and second articulated pitch angles of a wind tunnel sting drive system that will position a model near the centerline and at the desired angle of attack and angle of sideslip in the wind tunnel. These programs account for the effects of sting offset angles, sting bending angles and wind tunnel stream flow angles. In addition, the second program incorporates inputs from on board accelerometers that measure model pitch and roll with respect to gravity.

These program solve for the desired sting pitch and roll with an interactive procedure using the forward equations that calculate the model position, α and β from the sting geometry and the sting pitch and roll. This procedure allows sting offset angles, sting bending angles, and stream flow angles to be taken into account.

A copy of the source code of these two programs can be obtained from the Langley Computer Center with the following statements:

GET, ZSTNG2/UN = 003101N
or GET, ZTNG2PR/UN = 003101N

The run times for these programs vary depending upon the number of iterations required to converge to a solution. The programs were compiled and run under Fortran 5 (Fortran 77) on the Control Data Corporation Cyber CY180-860 computer at Langley Research Center. The run times for STNG2 are from 0.065 seconds for one iteration to 36.4 seconds for 100 iterations and 6 restarted calculations (the maximum allowed in these programs).

Normally, however, run time for STNG2 vary from about 1.8 seconds for large changes in α and β from the previous computation to 0.38 second for small changes in α and β (i.e., 2°). Very small changes (i.e., 0.01°) require only two iterations and 0.1 second run time.

Run times for STNG2PR are about twice as long as the run times for STNG2.

REFERENCES

1. Peterson, John B., Jr., Computer Programs for Calculation of Sting Pitch and Roll Angles Required to Obtain Angles of Attack and Sideslip on Wind Tunnel Models, NASA TM 100659, July 1988.
2. Heim, E. R. and Hobbs, R. W., Development of a Versatile Rotation Transformation Algorithm for Automatic Model Attitude Positioning. *Journal of Aircraft*, Vol. 26, No. 8, August 1989, pp. 718 - 722.
3. Letter Symbols for Aeronautical Sciences. ASA Y10.7-1954, The American Society of Mechanical Engineers, 154.
4. Foster, Jean M. and Adcock, Jerry B., Users Guide for the National Transonic Facility Data System, NASA TM-100511, December 1987.
5. Fox, Charles H., Jr., Real Time Reduction Capabilities at the Langley 7- by 10-Foot High Speed Tunnel, NASA TM-78801, January 1980.

APPENDIX A

COMPUTER LISTING OF PROGRAM STNG2

This appendix contains a computer listing of the program STNG2 which calculates the wind tunnel sting pitch and roll angles and second sting pitch angle required to obtain a desired angle of attack, α , and sideslip, β , on a wind tunnel model and to position the model in the center of the tunnel. The program accepts inputs of stream flow angles in two directions and sting offsets and sting bending angles in three directions

```

1          PROGRAM STNG2 (INPUT,OUTPUT)
2      C THIS PROGRAM CALCULATES THE STING ANGLES FOR THE HIGH ALPHA STING SYSTEM
3      C TO GET THE DESIRED ALPHA AND BETA AND THE MINIMUM DISTANCE FROM THE
4      C CENTERLINE OF THE TUNNEL ( ONLY THE SECOND PITCH ANGLE IS USED, NOT
5      C THE SECOND ROLL ANGLE)
6      C
7      C ANSWERS SAVED FOR NEXT COMPUTATION
8      C
9      C THE ORDER OF ROTATIONS ARE : STING PITCH (THESC), STING ROLL (PHISC),
10     C OFFSET YAW (PSIOFF), OFFSET PITCH (THEOFF), OFFSET ROLL (PHIOFF),
11     C SECOND STING PITCH (THESC2), SECOND STING ROLL (PHISC2),
12     C BENDING IN YAW (PSISB), BENDING IN PITCH (THESB), AND BENDING IN ROLL (PHISB)
13     C
14     C CODED BY -- JOHN B. PETERSON, JR.   NASA/LARC/AAD/HRNAB   1990
15     C
16
17     C          STING OFFSETS
18         PSIOFF=0.
19         THEOFF=-20.
20         PHIOFF=0.
21         PHISC2=0.
22
23     C          STING LENGTHS (SHOULD BE ON THE ORDER OF 57.3 UNITS SO THAT
24     C          DISTANCES AND ANGLES ARE OF SAME ORDER OF MAGNITUDE)
25     C          R1 IS THE RADIUS OF ROTATION OF THE ARC SECTOR.
26     C          X2 IS THE LENGTH OF THE MAIN STING.
27     C          X3 IS THE LENGTH OF THE SECOND (ARTICULATED) STING.
28     C
29         R1=122.5
30         X2=87.5
31         X3=48.918
32
33     C          LIMITS
34
35     C          LIMITS ON STING ROLL ANGLE
36     PHISLL=-85.

```

```

37      PHISLU=100.
38      C FOR INVERTED RUNS USE FOLLOWING LIMITS
39      C      PHISLL=80.
40      C      PHISLU=265.
41
42      C          LIMITS ON STING PITCH ANGLE (THESE LIMITS ALSO LIMIT
43      C          THE AVAILABLE BETA TO +19 & -11 UPRIGHT AND -19 & +11 INVERTED)
44      THESLL=-11.
45      THESLU=19.
46
47      C          STREAM FLOW ANGLES
48      C          POSITIVE FOR FLOW FROM BELOW AND FROM RIGHT
49      SWA=.0
50      UWA=.0
51      C          FREE STREAM VELOCITIES (TOTAL VEL. = 1.0)
52      U=SQRT( 1./ (1.+(TAND(SWA))**2+(TAND(UWA))**2 ) )
53      V=-U*TAND(SWA)
54      W= U*TAND(UWA)
55
56      C          SET INITIAL VALUES FOR VARIABLES
57
58      THESCUL=0.0
59      PHISCUL=0.0
60      THES2UL=20.
61
62      PHISCL=0.0
63      THESC2L=20.
64
65      THESSAV=999.
66      PHISSAV=999.
67      THES2SV=999.
68
69      C          TOLERANCES
70
71      TOLF=.00001
72      TOLDAB=.00001
73      TOLANG=.01
74

```

```

75          C          INPUTS TO CONTROL PROGRAM
76
77          C          COMMAND ANGLES
78          ALPC=40.
79          BETC=10.
80
81          C          STING DEFLECTIONS
82          PSISB=0.0
83          THESB=0.0
84          PHISB=0.0
85
86          C          END OF INPUTS TO CONTROL PROGRAM
87
88          C          SET TO PREVIOUS ANSWERS UNLIMITED IN THESC
89          THESC = THESCVL
90          PHISC = PHISCVL
91          THESC2 = THES2VL
92
93          C          SET REPEAT COUNTER TO 0
94          IREPT = 0
95
96          50 CONTINUE
97
98          IREPT = IREPT + 1
99
100         PRINT 98
101         PRINT 96
102         PRINT 99, THESC, PHISC, THESC2, PHISC2, ALPC, BETC, PSIOFF, THEOFF,
103         *          PHIOFF
104
105         C          CONVERGE ON ALPHA COMMAND (ALPC), BETA COMMAND (BETC),
106         C          AND DISTANCE (DIS)
107
108         DO 1000 ICONV=1,100
109
110         THESCSV=THESC-1.
111         CALL VEL (U,V,W, PSIOFF, THEOFF, PHIOFF, PSISB, THESB, PHISB,
112         *          THESCSV, PHISC, THESC2, PHISC2, ALPSV, BETSV)

```

```

113      CALL DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESCSV,PHISC,THESC2,
114      *      DYSV,DZSV)
115      DALPSV=ALPSV-ALPC
116      DBETSV=BETSV-BETC
117      DISSV = DZSV*COSD(PHISC) + DYSV*SIND(PHISC)
118      CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
119      *      THESC ,PHISC,THESC2,PHISC2,ALP ,BET )
120      CALL DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESC ,PHISC,THESC2,
121      *      DY ,DZ )
122      DALP  =ALP  - ALPC
123      DBET  =BET  - BETC
124      DIS   = DZ  *COSD(PHISC) + DY  *SIND(PHISC)
125      FA=ALP - ALPSV
126      FB=BET - BETSV
127      FD=DIS - DISSV
128      FSV = FA*DALPSV + FB*DBETSV + FD*DISSV
129      F   = FA*DALP  + FB*DBET  + FD*DIS
130      DO 100 ICTHE=1,3
131      CALL CONV(THESC ,F,THESCSV,FSV)
132      CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
133      *      THESC ,PHISC,THESC2,PHISC2,ALP ,BET )
134      CALL DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESC ,PHISC,THESC2,
135      *      DY ,DZ )
136      DALP  =ALP  - ALPC
137      DBET  =BET  - BETC
138      DIS   = DZ  *COSD(PHISC) + DY  *SIND(PHISC)
139      F     = FA*DALP  + FB*DBET  + FD*DIS
140      IF(ABS(F).LT.TOLF) GO TO 110
141      100 CONTINUE
142      110 CONTINUE
143      IF (THESC .EQ. 0.) THESC=.000001
144      PHISCSV=PHISC-1.
145      CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
146      *      THESC,PHISCSV,THESC2,PHISC2,ALPSV,BETSV)
147      CALL DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESC,PHISCSV,THESC2,
148      *      DYSV,DZSV)
149      DALPSV=ALPSV-ALPC
150      DBETSV=BETSV-BETC

```

```

151      DISSV = DZSV*COSD(PHISCSV) + DYSV*SIND(PHISCSV)
152      CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESEB,PHISB,
153      *          THESC,PHISC ,THESC2,PHISC2,ALP ,BET )
154      CALL DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESC,PHISC ,THESC2,
155      *          DY ,DZ )
156      DALP=ALP-ALPC
157      DBET=BET-BETC
158      DIS  = DZ *COSD(PHISC) + DY *SIND(PHISC)
159      FA=ALP-ALPSV
160      FB=BET-BETSV
161      FD=DIS - DISSV
162      FSV = FA*DALPSV + FB*DBETSV + FD*DISSV
163      F   = FA*DALP  + FB*DBET  + FD*DIS
164      DO 200 ICPHI=1,3
165      CALL CONV(PHISC,F,PHISCSV,FSV)
166      CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESEB,PHISB,
167      *          THESC,PHISC ,THESC2,PHISC2,ALP ,BET )
168      CALL DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESC,PHISC ,THESC2,
169      *          DY ,DZ )
170      DALP=ALP-ALPC
171      DBET=BET-BETC
172      DIS  = DZ *COSD(PHISC) + DY *SIND(PHISC)
173      F   = FA*DALP  + FB*DBET  + FD*DIS
174      IF(ABS(F).LT.TOLF) GO TO 210
175      200 CONTINUE
176      210 CONTINUE
177      THESC2S=THESC2-1.
178      CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESEB,PHISB,
179      *          THESC,PHISC,THESC2S,PHISC2,ALPSV,BETSV)
180      CALL DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESC,PHISC,THESC2S,
181      *          DYSV,DZSV)
182      DALPSV=ALPSV-ALPC
183      DBETSV=BETSV-BETC
184      DISSV = DZSV*COSD(PHISC) + DYSV*SIND(PHISC)
185      CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESEB,PHISB,
186      *          THESC,PHISC,THESC2 ,PHISC2,ALP ,BET )
187      CALL DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESC,PHISC,THESC2 ,
188      *          DY ,DZ )

```

```

189      DALP=ALP-ALPC
190      DBET=BET-BETC
191      DIS  = DZ *COSD(PHISC) + DY *SIND(PHISC)
192      FA=ALP -ALPSV
193      FB=BET -BETSV
194      FD=DIS - DISSV
195      FSV = FA*DALPSV + FB*DBETSV + FD*DISSV
196      F   = FA*DALP  + FB*DBET  + FD*DIS
197      DO 300 ICTHE2=1,3
198      CALL CONV(THESC2,F,THESC2S,FSV)
199      CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
200      *          THESC,PHISC,THESC2 ,PHISC2,ALP ,BET )
201      CALL DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESC,PHISC,THESC2 ,
202      *          DY ,DZ )
203      DALP=ALP-ALPC
204      DBET=BET-BETC
205      DIS  = DZ *COSD(PHISC) + DY *SIND(PHISC)
206      F   = FA*DALP  + FB*DBET  + FD*DIS
207      IF(ABS(F).LT.TOLF) GO TO 310
208      300 CONTINUE
209      310 CONTINUE
210
211      C          CHECK TO SEE IF MOVEMENT IN THESC, PHISC AND THESC2 IS LESS THAN
212      C          TOLANG IN LAST ITERATION
213      IF ( ABS(THESC-THESAV).LT.TOLANG
214      * .AND. ABS(PHISC-PHISSAV).LT.TOLANG
215      * .AND. ABS(THESC2-THES2SV).LT.TOLANG) GO TO 1010
216      THESAV=THESC
217      PHISSAV=PHISC
218      THES2SV=THESC2
219      1000 CONTINUE
220      1010 CONTINUE
221
222      PRINT 97, ICONV
223      PRINT 94
224      PRINT 99, THESC, PHISC, THESC2, PHISC2,ALP, BET,DY,DZ
225
226      IF (IREPT.GE.12) GO TO 3020

```

```

227
228           C           CHECK TO SEE IF CONVERGED OUTSIDE PHIS LIMIT
229           IF(PHISC.GT.PHISLU) THEN
230             PHISC=PHISC-180.
231             THESC=-THESC
232             GO TO 50
233           END IF
234           IF(PHISC.LT.PHISLL) THEN
235             PHISC=PHISC+180.
236             THESC=-THESC
237             GO TO 50
238           END IF
239
240           C           SAVE ANSWERS UNLIMITED IN PITCH FOR NEXT COMPUTATION
241             THESCUL=THESC
242             PHISCUL=PHISC
243             THES2UL=THESC2
244
245           C           CHECK TO SEE IF THESC CONVERGED OUTSIDE THESC LIMIT
246
247           IF (THESC.LT.THESLL) THEN
248             THESC=THESLL
249             GO TO 2020
250           END IF
251           IF (THESC.GT.THESLU) THEN
252             THESC=THESLU
253             GO TO 2020
254           END IF
255
256             PRINT 98
257             PRINT 96
258             PRINT 99,THESC,PHISC,THESC2,PHISC2,ALPC,BETC,PSIOFF,THEOFF,
259           *             PHIOFF
260
261           C           FINAL CONVERGENCE ON ALPHA COMMAND (ALPC) AND BETA COMMAND (BETC)
262           C           WITHOUT REGARD TO DISTANCE (DIS)
263
264           DO 2000 ICONV=1,100

```

```

265
266      THESCSV=THESC-1.
267      CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
268      *          THESCSV,PHISC,THESC2,PHISC2,ALPSV,BETSV)
269      DALPSV=ALPSV-ALPC
270      DBETSV=BETSV-BETC
271      CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
272      *          THESC ,PHISC,THESC2,PHISC2,ALP ,BET )
273      DALP  =ALP  - ALPC
274      DBET  =BET  - BETC
275      FA=ALP - ALPSV
276      FB=BET - BETSV
277      FSV = FA*DALPSV + FB*DBETSV
278      F   = FA*DALP  + FB*DBET
279      DO 1100 ICTHE=1,3
280      CALL CONV(THESC ,F,THESCSV,FSV)
281      CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
282      *          THESC ,PHISC,THESC2,PHISC2,ALP ,BET )
283      DALP  =ALP  - ALPC
284      DBET  =BET  - BETC
285      F   = FA*DALP  + FB*DBET
286      IF(ABS(F).LT.TOLF) GO TO 1110
287      1100 CONTINUE
288      1110 CONTINUE
289      IF(THESC.EQ.0.) THESC=.000001
290      PHISCSV=PHISC-1.
291      CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
292      *          THESC,PHISCSV,THESC2,PHISC2,ALPSV,BETSV)
293      DALPSV=ALPSV-ALPC
294      DBETSV=BETSV-BETC
295      CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
296      *          THESC,PHISC ,THESC2,PHISC2,ALP ,BET )
297      DALP=ALP-ALPC
298      DBET=BET-BETC
299      FA=ALP-ALPSV
300      FB=BET-BETSV
301      FSV = FA*DALPSV + FB*DBETSV
302      F   = FA*DALP  + FB*DBET

```

```

303          DO 1200 ICPHI=1,3
304          CALL CONV(PHISC,F,PHISCSV,FSV)
305          CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
306          *          THESC,PHISC ,THESC2,PHISC2,ALP ,BET )
307          DALP=ALP-ALPC
308          DBET=BET-BETC
309          F   = FA*DALP  + FB*DBET
310          IF(ABS(F).LT.TOLF) GO TO 1210
311          1200 CONTINUE
312          1210 CONTINUE
313          THESC2S=THESC2-1.
314          CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
315          *          THESC,PHISC,THESC2S,PHISC2,ALPSV,BETSV)
316          DALPSV=ALPSV-ALPC
317          DBETSV=BETSV-BETC
318          CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
319          *          THESC,PHISC,THESC2 ,PHISC2,ALP ,BET )
320          DALP=ALP-ALPC
321          DBET=BET-BETC
322          FA=ALP -ALPSV
323          FB=BET -BETSV
324          FSV = FA*DALPSV + FB*DBETSV
325          F   = FA*DALP  + FB*DBET
326          DO 1300 ICTHE2=1,3
327          CALL CONV(THESC2,F,THESC2S,FSV)
328          CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
329          *          THESC,PHISC,THESC2 ,PHISC2,ALP ,BET )
330          DALP=ALP-ALPC
331          DBET=BET-BETC
332          F   = FA*DALP  + FB*DBET
333          IF(ABS(F).LT.TOLF) GO TO 1310
334          1300 CONTINUE
335          1310 CONTINUE
336
337          C          CHECK TO SEE IF ALP AND BET ARE WITHIN TOLDAB OF ALPC AND BETC
338          IF ( (ABS(DALP)+ABS(DBET)) .LT. TOLDAB ) GO TO 2010
339
340          2000 CONTINUE

```

```

341      2010 CONTINUE
342
343      PRINT 97, ICONV
344      PRINT 94
345      PRINT 99, THESC, PHISC, THESC2, PHISC2,ALP, BET
346
347      IF (IREPT.GE.6) GO TO 3020
348
349      C          CHECK TO SEE IF THESC CONVERGED OUTSIDE THESC LIMIT
350
351      IF (THESC.LT.THESLL) THEN
352          THESC=THESLL
353          GO TO 2020
354      END IF
355      IF (THESC.GT.THESLU) THEN
356          THESC=THESLU
357          GO TO 2020
358      END IF
359
360      C          CHECK TO SEE IF CONVERGED OUTSIDE PHIS LIMIT
361      IF(PHISC.GT.PHISLU) THEN
362          PHISC=PHISC - 180.
363          THESC= -THESC
364          GO TO 50
365      END IF
366      IF(PHISC.LT.PHISLL) THEN
367          PHISC=PHISC + 180.
368          THESC= -THESC
369          GO TO 50
370      END IF
371
372      C          GO TO END
373      GO TO 3020
374
375      2020 CONTINUE
376      C          FINAL CONVERGENCE ON ALPHA COMMAND (ALPC) AND BETA COMMAND (BETC)
377      C          WITHOUT REGARD TO DISTANCE (DIS), AND WITH THESC SET TO
378      C          THE LIMITS (THESLL OR THESLU)

```

```

379
380 C          SET TO PREVIOUS ANSWERS LIMITED IN THE SC
381 PHISC = PHISCL
382 THE SC2 = THE SC2L
383
384          PRINT 98
385          PRINT 96
386          PRINT 99, THE SC, PHISC, THE SC2, PHISC2, ALPC, BETC, PSIOFF, THE OFF,
387 *          PHIOFF
388
389          DO 3000 ICONV=1,100
390
391          THE SC2S=THE SC2-1.
392          CALL VEL (U,V,W,PSIOFF,THE OFF,PHIOFF,PSISB,THE SB,PHISB,
393 *          THE SC,PHISC,THE SC2S,PHISC2,ALPSV,BETSV)
394          DALPSV=ALPSV-ALPC
395          DBETSV=BETSV-BETC
396          CALL VEL (U,V,W,PSIOFF,THE OFF,PHIOFF,PSISB,THE SB,PHISB,
397 *          THE SC,PHISC,THE SC2 ,PHISC2,ALP ,BET )
398          DALP=ALP-ALPC
399          DBET=BET-BETC
400          FA=ALP -ALPSV
401          FB=BET -BETSV
402          FSV = FA*DALPSV + FB*DBETSV
403          F   = FA*DALP  + FB*DBET
404          DO 2300 ICTHE2=1,3
405          CALL CONV(THE SC2,F,THE SC2S,FSV)
406          CALL VEL (U,V,W,PSIOFF,THE OFF,PHIOFF,PSISB,THE SB,PHISB,
407 *          THE SC,PHISC,THE SC2 ,PHISC2,ALP ,BET )
408          DALP=ALP-ALPC
409          DBET=BET-BETC
410          F   = FA*DALP  + FB*DBET
411          IF(ABS(F).LT.TOLF) GO TO 2310
412          2300 CONTINUE
413          2310 CONTINUE
414
415          IF(THE SC.EQ.0.) THE SC=.000001
416          PHISCSV=PHISC-1.

```

```

417      CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
418      *          THESC,PHISCSV,THESC2,PHISC2,ALPSV,BETSV)
419      DALPSV=ALPSV-ALPC
420      DBETSV=BETSV-BETC
421      CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
422      *          THESC,PHISC ,THESC2,PHISC2,ALP ,BET )
423      DALP=ALP-ALPC
424      DBET=BET-BETC
425      FA=ALP-ALPSV
426      FB=BET-BETSV
427      FSV = FA*DALPSV + FB*DBETSV
428      F   = FA*DALP  + FB*DBET
429      DO 2200 ICPHI=1,3
430      CALL CONV(PHISC,F,PHISCSV,FSV)
431      CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
432      *          THESC,PHISC ,THESC2,PHISC2,ALP ,BET )
433      DALP=ALP-ALPC
434      DBET=BET-BETC
435      F   = FA*DALP  + FB*DBET
436      IF(ABS(F).LT.TOLF) GO TO 2210
437      2200 CONTINUE
438      2210 CONTINUE
439      C          CHECK TO SEE IF ALP AND BET ARE WITHIN TOLDAB OF ALPC AND BETC
440      IF ( (ABS(DALP)+ABS(DBET)) .LT. TOLDAB ) GO TO 3010
441
442      3000 CONTINUE
443      3010 CONTINUE
444
445      PRINT 97, ICONV
446      PRINT 94
447      PRINT 99, THESC, PHISC, THESC2, PHISC2,ALP, BET
448
449      IF (IREPT.GE.6) GO TO 3020
450
451      C          CHECK TO SEE IF CONVERGED OUTSIDE THIS LIMIT AND SAVE ANSWERS
452      C          WITH PITCH LIMITED TO THESCLL OR THESCLU FOR NEXT COMPUTATION
453      IF(PHISC.GT.PHISLU) THEN
454      PHISC=PHISC - 180.

```

```

455         THESC= -THESC
456         PHISCL=PHISC
457         THESC2L=THESC2
458         GO TO 50
459     END IF
460     IF(PHISC.LT.PHISLL) THEN
461         PHISC=PHISC + 180.
462         THESC= -THESC
463         PHISCL=PHISC
464         THESC2L=THESC2
465         GO TO 50
466     END IF
467
468     C             SAVE ANSWERS WITH PITCH LIMITED TO THESLL OR THESLU
469     C             FOR NEXT COMPUTATION
470         PHISCL=PHISC
471         THESC2L=THESC2
472     3020 CONTINUE
473         CALL DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESC,PHISC,THESC2 ,
474         *         DY ,DZ )
475         PRINT 98
476         PRINT 94
477         PRINT 99, THESC, PHISC, THESC2, PHISC2,ALP, BET,DY,DZ
478         PRINT 98
479         PRINT 98
480     STOP
481     99     FORMAT (6F9.4,2F7.3,F10.5)
482     98     FORMAT ( )
483     97     FORMAT(1H ,"ICONV =",I3)
484     96     FORMAT("  THESC    PHISC    THESC2    PHISC2    ALPC    BETC",
485     *         "  PSIOFF  THEOFF  PHIOFF")
486     95     FORMAT("  THESC    PHISC    THESC2    PHISC2    DALP",
487     *         "  DBET     DY      DZ      F")
488     94     FORMAT("  THESC    PHISC    THESC2    PHISC2    ALP    BET ",
489     *         "  DY      DZ")
490     93     FORMAT (6F9.4,14X,F10.5)
491     END

```

```

1          SUBROUTINE VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
2          *          THESC,PHISC,THESC2,PHISC2,ALP,BET)
3          C
4          C THIS SUBROUTINE CALCULATES THE ANGLE OF ATTACK (ALP) AND ANGLE OF
5          C SIDESLIP (BET) OF A WIND TUNNEL MODEL.  INPUTS ARE VELOCITY COMPONENTS
6          C IN THE WIND TUNNEL (U, V, AND W), THE STING OFFSET ANGLES (PSIOFF, THEOFF,
7          C AND PHIOFF), THE STING BENDING ANGLES (PSISB, THESB, AND PHISB, AND
8          C THE STING DRIVE ANGLES (THESC, PHISC, THESC2 AND PHISC2).  THE SUBROUTINE
9          C CALCULATES THE COMPONENTS OF THE FREE STREAM VELOCITY ALONG THE THREE
10         C AXES OF THE MODEL AFTER EACH ROTATION.  AFTER THE LAST ROTATION, THESE
11         C VELOCITIES ARE USED TO CALCULATE THE ANGLE OF ATTACK (ALP) AND THE
12         C ANGLE OF SIDESLIP (BET).
13         C
14         DOR=57.2957795
15         C          STING PITCH (Y)
16         UB=U*COSD(THESC)-W*SIND(THESC)
17         VB=V
18         WB=W*COSD(THESC)+U*SIND(THESC)
19         C          STING ROLL (X)
20         UC=UB
21         VC=VB*COSD(PHISC)-WB*SIND(PHISC)
22         WC=WB*COSD(PHISC)+VB*SIND(PHISC)
23         C          OFFSET YAW (Z)
24         UD=UC*COSD(PSIOFF)-VC*SIND(PSIOFF)
25         VD=VC*COSD(PSIOFF)+UC*SIND(PSIOFF)
26         WD=WC
27         C          OFFSET PITCH (Y)
28         UE=UD*COSD(THEOFF)-WD*SIND(THEOFF)
29         VE=VD
30         WE=WD*COSD(THEOFF)+UD*SIND(THEOFF)
31         C          OFFSET ROLL (X)
32         UF=UE
33         VF=VE*COSD(PHIOFF)-WE*SIND(PHIOFF)
34         WF=WE*COSD(PHIOFF)+VE*SIND(PHIOFF)
35         C          2ND STING PITCH (Y)
36         UG=UF*COSD(THESC2)-WF*SIND(THESC2)

```

```

37      VG=VF
38      WG=WF*COSD(THESC2)+UF*SIND(THESC2)
39      C      2ND STING ROLL (X)
40      UH=UG
41      VH=VG*COSD(PHISC2)-WG*SIND(PHISC2)
42      WH=WG*COSD(PHISC2)+VG*SIND(PHISC2)
43      C      STING BENDING IN YAW (Z)
44      UI=UH*COSD(PSISB)-VH*SIND(PSISB)
45      VI=VH*COSD(PSISB)+UH*SIND(PSISB)
46      WI=WH
47      C      STING BENDING IN PITCH (Y)
48      UJ=UI*COSD(THESB)-WI*SIND(THESB)
49      VJ=VI
50      WJ=WI*COSD(THESB)+UI*SIND(THESB)
51      C      STING BENDING IN ROLL (X)
52      UK=UJ
53      VK=VJ*COSD(PHISB)-WJ*SIND(PHISB)
54      WK=WJ*COSD(PHISB)+VJ*SIND(PHISB)
55      C      ALPHA AND BETA
56      IF(WK.EQ.0..AND.UK.EQ.0.)UK=.0000001
57      ALP=ATAN2(WK,UK)*DOR
58      IF(VK.LT.-1.)VK=-1.
59      IF(VK.GT.1.)VK=1.
60      BET=ASIN(-VK)*DOR
61      RETURN
62      END

```

```

1          SUBROUTINE DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESC,PHISC,THESC2,
2          *          DY,DZ)
3          C
4          C      THIS SUBROUTINE CALCULATES THE POSITION OF THE MODEL CENTER
5          C WITH RESPECT TO THE TUNNEL AXIS SYSTEM.  THE SUBROUTINE STARTS AT
6          C THE MODEL AND PROCEEDS TO THE TUNNEL AXIS SYSTEM.  THE ROTATIONS
7          C AND TRANSLATIONS ARE TAKEN IN REVERSE ORDER FROM THE ORDER USED
8          C IN THE MAIN PROGRAM.  THE STING BENDING ANGLES ARE IGNORED.
9          C THE ORDER OF ROTATIONS ARE:
10         C SECOND STING ROLL (PHISC2), SECOND STING PITCH (THESC2),
11         C OFFSET ROLL (PSIOFF), OFFSET PITCH (THEOFF), OFFSET YAW (PSIOFF),
12         C STING ROLL (PHISC), AND STING PITCH (THESC).
13         C          R1 IS THE RADIUS OF ROTATION OF THE ARC SECTOR.
14         C          X2 IS THE LENGTH OF THE MAIN STING.
15         C          X3 IS THE LENGTH OF THE SECOND (ARTICULATED) STING.
16         C
17         C          XA=0.
18         C          YA=0.
19         C          ZA=0.
20
21         C          TRANSLATION (X)  HIGH ALPHA STING
22         C          XB=XA - X3
23         C          YB=YA
24         C          ZB=ZA
25
26         C          2ND STING ROLL (X)
27         C          NO EFFECT ON X, Y OR Z
28
29         C          2ND STING PITCH (Y)
30         C          XC=XB*COSD(-THESC2)-ZB*SIND(-THESC2)
31         C          YC=YB
32         C          ZC=ZB*COSD(-THESC2)+XB*SIND(-THESC2)
33
34         C          TRANSLATION (X)  MAIN STING
35         C          XD=XC - X2
36         C          YD=YC

```

```

37         ZD=ZC
38
39         C         OFFSET ROLL (X)
40         XE=XD
41         YE=YD*COSED(-PHIOFF)-ZD*SIND(-PHIOFF)
42         ZE=ZD*COSED(-PHIOFF)+YD*SIND(-PHIOFF)
43
44         C         OFFSET PITCH (Y)
45         XF=XE*COSED(-THEOFF)-ZE*SIND(-THEOFF)
46         YF=YE
47         ZF=ZE*COSED(-THEOFF)+XE*SIND(-THEOFF)
48
49         C         OFFSET YAW (Z)
50         XG=XF*COSED(-PSIOFF)-YF*SIND(-PSIOFF)
51         YG=YF*COSED(-PSIOFF)+XF*SIND(-PSIOFF)
52         ZG=ZF
53
54         C         TRANSLATION (X) ,ARC SECTOR
55         XH=XG + R1
56         YH=YG
57         ZH=ZG
58
59         C         STING ROLL (X)
60         XI=XH
61         YI=YH*COSED(-PHISC)-ZH*SIND(-PHISC)
62         ZI=ZH*COSED(-PHISC)+YH*SIND(-PHISC)
63
64         C         STING PITCH (Y)
65         XJ=XI*COSED(-THESC)-ZI*SIND(-THESC)
66         YJ=YI
67         ZJ=ZI*COSED(-THESC)+XI*SIND(-THESC)
68
69         DY=YJ
70         DZ=ZJ
71
72         RETURN
73         END

```

```

1          SUBROUTINE CONV (X,Y,XSAVE,YSAVE)
2
3          C
4          C THIS SUBROUTINE CALCULATES THE VALUE OF X WHICH MINIMIZES THE VALUE OF Y.
5          C THE CALLING PROGRAM CALCULATES THE VALUES OF Y FOR EACH VALUE OF X.
6          C THE NEW VALUE OF X IS DETERMINED BY CALCULATING WHERE A STRAIGHT LINE
7          C THROUGH THE LAST TWO PREVIOUS PAIR OF POINTS, X AND Y, AND XSAVE AND YSAVE,
8          C INTERSECT THE X AXIS. THIS NEW VALUE OF X IS RETURNED TO THE CALLING
9          C PROGRAM IN THE X PARAMETER.
10         X1=XSAVE
11         Y1=YSAVE
12         X2=X
13         Y2=Y
14         IF(X2.EQ.X1) GO TO 3
15         IF(Y2.EQ.Y1) GO TO 4
16         XK=(Y2-Y1)/(X2-X1)
17         X=X2-Y2/XK
18         XA=(X1+X2)/2.
19         DX=ABS(X2-X1)
20         IF(ABS(X-XA).GT.3.5*DX) X=XA+3.5*SIGN( DX, (X-XA) )
21         GO TO 100
22         3 X=X2-Y2
23         GO TO 100
24         4 X=(X2+X1)/2.
25         100 XSAVE=X2
26         YSAVE=Y2
27         IF(ABS(Y1).LT.ABS(Y2)) THEN
28             XSAVE=X1
29             YSAVE=Y1
30         END IF
31         RETURN
32         END

```

APPENDIX B

COMPUTER LISTING OF PROGRAM STNG2PR

This appendix contains a computer listing of the program STNG2PR which calculates the wind tunnel sting pitch and roll angles and second sting pitch angle required to obtain a desired angle of attack, α , and sideslip, β , and to position the model in the center of the tunnel for a wind tunnel model with on board accelerometers to measure model pitch and/or roll angles . The program accepts accelerometer measurements of model pitch and roll, wind tunnel stream flow angles in two directions and sting offsets and sting bending angles in three directions

```

1          PROGRAM STNG2PR (INPUT,OUTPUT)
2          C THIS PROGRAM CALCULATES THE STING ANGLES FOR THE HIGH ALPHA STING SYSTEM
3          C TO GET THE DESIRED ALPHA AND BETA AND THE MINIMUM DISTANCE FROM THE
4          C CENTERLINE OF THE TUNNEL ( ONLY THE SECOND PITCH ANGLE IS USED, NOT
5          C THE SECOND ROLL ANGLE)
6          C THE PROGRAM ACCEPTS INPUTS FROM ACCELEROMETERS ON BOARD THE MODEL TO
7          C MEASURE THE MODEL PITCH AND ROLL RELATIVE TO GRAVITY.
8          C
9          C ANSWERS SAVED FOR NEXT COMPUTATION
10         C
11         C THE ORDER OF ROTATIONS ARE : STING PITCH (THESC), STING ROLL (PHISC),
12         C OFFSET YAW (PSIOFF), OFFSET PITCH (THEOFF), OFFSET ROLL (PHIOFF),
13         C SECOND STING PITCH (THESC2), SECOND STING ROLL (PHISC2),
14         C BENDING IN YAW (PSISB), BENDING IN PITCH (THESB), AND BENDING IN ROLL (PHISB)
15         C
16         C CODED BY -- JOHN B. PETERSON, JR.   NASA/LARC/AAD/HRNAB   1990
17         C
18
19         C          STING OFFSETS
20         C          PSIOFF=0.
21         C          THEOFF=-20.
22         C          PHIOFF=0.
23         C          PHISC2=0.
24
25         C          STING LENGTHS (SHOULD BE ON THE ORDER OF 57.3 UNITS SO THAT
26         C          DISTANCES AND ANGLES ARE OF SAME ORDER OF MAGNITUDE)
27         C          R1 IS THE RADIUS OF ROTATION OF THE ARC SECTOR.
28         C          X2 IS THE LENGTH OF THE MAIN STING.
29         C          X3 IS THE LENGTH OF THE SECOND (ARTICULATED) STING.
30         C
31         C          R1=122.5
32         C          X2=87.5
33         C          X3=48.918
34
35         C          LIMITS
36

```

```

37      C          LIMITS ON STING ROLL ANGLE
38      PHISLL=-85.
39      PHISLU=100.
40      C FOR INVERTED RUNS USE FOLLOWING LIMITS
41      C          PHISLL=80.
42      C          PHISLU=265.
43
44      C          LIMITS ON STING PITCH ANGLE (THESE LIMITS ALSO LIMIT
45      C          THE AVAILABLE BETA TO +19 & -11 UPRIGHT AND -19 & +11 INVERTED)
46      THESLL=-11.
47      THESLU=19.
48
49      C          STREAM FLOW ANGLES
50      C          POSITIVE FOR FLOW FROM BELOW AND FROM RIGHT
51      SWA=.0
52      UWA=.0
53      C          FREE STREAM VELOCITIES (TOTAL VEL. = 1.0)
54      U=SQRT( 1./ (1.+(TAND(SWA))**2+(TAND(UWA))**2 ) )
55      V=-U*TAND(SWA)
56      W= U*TAND(UWA)
57
58      C          SET INITIAL VALUES FOR VARIABLES
59      THESC = 0.0
60      PHISC = 0.0
61      THESC2 = 20.
62      PHISC2 = 0.0
63
64      THESCU=0.0
65      PHISCU=0.0
66      THES2U=20.
67
68      PHISCL=0.0
69      THESC2L=20.
70
71      THESSAV=999.
72      PHISSAV=999.
73      THES2SV=999.
74

```

```

75      C          TOLERANCES
76      TOLF=.00001
77      TOLDAB=.00001
78      TOLANG=.01
79
80      C          INPUTS TO CONTROL PROGRAM
81
82      C          COMMAND ANGLES
83      ALPC=40.
84      BETC=10.
85
86      C          STING DEFLECTIONS
87      PSISB=0.0
88      THESB=0.0
89      PHISB=0.0
90
91      C          STING POSITION
92      THES=THESC
93      PHIS=PHISC
94      THES2=THESC2
95      PHIS2=PHISC2
96
97      C          MODEL PITCH AND ROLL (SUBROUTINE SIMUST IS USED HERE
98      C          TO GENERATE VALUES OF MODEL PITCH AND ROLL FOR THIS PROGRAM.
99      C          ACTUALLY THEY WILL BE MEASURED WITH ACCELEROMETERS ON BOARD
100     C          THE MODEL AND INPUT INTO THE PROGRAM AS THEM03 AND PHIM03)
101     C
102     CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
103     *          THES ,PHIS ,THES2 ,PHIS2,PSIM03,THEM03,PHIM03)
104     THEM03=THEM03+0.
105     PHIM03=PHIM03+0.
106
107     C          END OF INPUTS TO CONTROL PROGRAM
108
109     C          SET TO PREVIOUS ANSWERS UNLIMITED IN THESC
110     THESC = THESCUL
111     PHISC = PHISCUL
112     THESC2 = THESC2UL

```

```

113
114 C          START OF CONTROL PROGRAM
115
116 C          DETERMINE THEORETICAL PITCH AND ROLL OF MODEL
117 C          AND COMPARE WITH MEASURED PITCH AND ROLL TO GET ERROR
118
119          CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESEB,PHISB,
120 *          THES ,PHIS ,THES2 ,PHIS2,PSIMT,THEMT,PHIMT)
121          DTHEMOB=THEMOB-THEMT
122          DPHIMOB=PHIMOB-PHIMT
123
124 *****
125 C IF THEMOB OR PHIMOB IS NOT AVAILABLE, DTHEMOB OR DPHIMOB
126 C SHOULD BE SET TO ZERO
127 C     DTHEMOB=0.
128 C     DPHIMOB=0.
129 *****
130
131          PRINT 98
132          PRINT 92
133          PRINT 99,THEMOB,PHIMOB,DTHEMOB,DPHIMOB
134
135 C          SET REPEAT COUNTER TO 0
136 C          IREPT = 0
137
138 C          START OVER POINT IF THE PROGRAM CONVERGES OUTSIDE THE
139 C          THE STING ROLL LIMITS (PHISLL & PHISLU)
140 C          50 CONTINUE
141
142          IREPT = IREPT + 1
143
144          PRINT 98
145          PRINT 96
146          PRINT 99,THESEB,PHISB,THESEB2,PHISB2,ALPC,BETC,PSIOFF,THEOFF,
147 *          PHIOFF
148
149 C          CONVERGE ON ALPHA COMMAND (ALPC), BETA COMMAND (BETC),
150 C          AND DISTANCE (DIS)

```

```

151
152 DO 1000 ICONV=1,100
153
154 THESCSV=THESC-1.
155 CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
156 * THESCSV,PHISC ,THESC2 ,PHISC2,PSIMT,THEMT,PHIMT)
157 THEMObT=THEMT+DTHEMOB
158 PHIMObT=PHIMT+DPHIMOB
159 CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMObT,ALPSV,BETSV)
160 CALL DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESCSV,PHISC,THESC2,
161 * DYSV,DZSV)
162 DALPSV=ALPSV-ALPC
163 DBETSV=BETSV-BETC
164 DISSV = DZSV*COSD(PHISC) + DYSV*SIND(PHISC)
165 CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
166 * THESC ,PHISC ,THESC2 ,PHISC2,PSIMT,THEMT,PHIMT)
167 THEMObT=THEMT+DTHEMOB
168 PHIMObT=PHIMT+DPHIMOB
169 CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMObT,ALP ,BET )
170 CALL DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESC ,PHISC,THESC2,
171 * DY ,DZ )
172 DALP =ALP - ALPC
173 DBET =BET - BETC
174 DIS = DZ *COSD(PHISC) + DY *SIND(PHISC)
175 FA=ALP - ALPSV
176 FB=BET - BETSV
177 FD=DIS - DISSV
178 FSV = FA*DALPSV + FB*DBETSV + FD*DISSV
179 F = FA*DALP + FB*DBET + FD*DIS
180 DO 100 ICTHE=1,3
181 CALL CONV(THESC ,F,THESCSV,FSV)
182 CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
183 * THESC ,PHISC ,THESC2 ,PHISC2,PSIMT,THEMT,PHIMT)
184 THEMObT=THEMT+DTHEMOB
185 PHIMObT=PHIMT+DPHIMOB
186 CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMObT,ALP ,BET )
187 CALL DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESC ,PHISC,THESC2,
188 * DY ,DZ )

```

```

189      DALP =ALP - ALPC
190      DBET =BET - BETC
191      DIS  = DZ *COSD(PHISC) + DY *SIND(PHISC)
192      F    = FA*DALP  + FB*DBET  + FD*DIS
193      IF(ABS(F).LT.TOLF) GO TO 110
194      100 CONTINUE
195      110 CONTINUE
196      IF (THESC .EQ. 0.) THESC=.000001
197      PHISCSV=PHISC-1.
198      CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
199      *          THESC ,PHISCSV,THESC2 ,PHISC2,PSIMT,THEMT,PHIMT)
200      THEMOBT=THEMT+DTHEMOB
201      PHIMOBT=PHIMT+DPHIMOB
202      CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALPSV,BETSV)
203      CALL DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESC,PHISCSV,THESC2,
204      *          DYSV,DZSV)
205      DALPSV=ALPSV-ALPC
206      DBETSV=BETSV-BETC
207      DISSV = DZSV*COSD(PHISCSV) + DYSV*SIND(PHISCSV)
208      CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
209      *          THESC ,PHISC ,THESC2 ,PHISC2,PSIMT,THEMT,PHIMT)
210      THEMOBT=THEMT+DTHEMOB
211      PHIMOBT=PHIMT+DPHIMOB
212      CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALP ,BET )
213      CALL DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESC,PHISC ,THESC2,
214      *          DY ,DZ )
215      DALP=ALP-ALPC
216      DBET=BET-BETC
217      DIS  = DZ *COSD(PHISC) + DY *SIND(PHISC)
218      FA=ALP-ALPSV
219      FB=BET-BETSV
220      FD=DIS - DISSV
221      FSV = FA*DALPSV + FB*DBETSV + FD*DISSV
222      F    = FA*DALP  + FB*DBET  + FD*DIS
223      DO 200 ICPHI=1,3
224      CALL CONV(PHISC,F,PHISCSV,FSV)
225      CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
226      *          THESC ,PHISC ,THESC2 ,PHISC2,PSIMT,THEMT,PHIMT)

```

```

227      THEMOBT=THEMT+DTHEMOB
228      PHIMOBT=PHIMT+DPHIMOB
229      CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALP ,BET )
230      CALL DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESC,PHISC ,THESC2,
231      *      DY ,DZ )
232      DALP=ALP-ALPC
233      DBET=BET-BETC
234      DIS = DZ *COSD(PHISC) + DY *SIND(PHISC)
235      F = FA*DALP + FB*DBET + FD*DIS
236      IF(ABS(F).LT.TOLF) GO TO 210
237 200 CONTINUE
238 210 CONTINUE
239      THESC2S=THESC2-1.
240      CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
241      *      THESC ,PHISC ,THESC2S,PHISC2,PSIMT,THEMT,PHIMT)
242      THEMOBT=THEMT+DTHEMOB
243      PHIMOBT=PHIMT+DPHIMOB
244      CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALPSV,BETSV)
245      CALL DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESC,PHISC,THESC2S,
246      *      DYSV,DZSV)
247      DALPSV=ALPSV-ALPC
248      DBETSV=BETSV-BETC
249      DISSV = DZSV*COSD(PHISC) + DYSV*SIND(PHISC)
250      CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
251      *      THESC ,PHISC ,THESC2 ,PHISC2,PSIMT,THEMT,PHIMT)
252      THEMOBT=THEMT+DTHEMOB
253      PHIMOBT=PHIMT+DPHIMOB
254      CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALP ,BET )
255      CALL DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESC,PHISC,THESC2 ,
256      *      DY ,DZ )
257      DALP=ALP-ALPC
258      DBET=BET-BETC
259      DIS = DZ *COSD(PHISC) + DY *SIND(PHISC)
260      FA=ALP -ALPSV
261      FB=BET -BETSV
262      FD=DIS - DISSV
263      FSV = FA*DALPSV + FB*DBETSV + FD*DISSV
264      F = FA*DALP + FB*DBET + FD*DIS

```

```

265      DO 300 ICTHE2=1,3
266      CALL CONV(THESC2,F,THESC2S,FSV)
267      CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
268      *          THESC ,PHISC ,THESC2 ,PHISC2,PSIMT,THEMT,PHINT)
269      THEMOBT=THEMT+DTHEMOB
270      PHIMOBT=PHINT+DPHIMOB
271      CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALP ,BET )
272      CALL DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESC,PHISC,THESC2 ,
273      *          DY ,DZ )
274      DALP=ALP-ALPC
275      DBET=BET-BETC
276      DIS  = DZ *COSD(PHISC) + DY *SIND(PHISC)
277      F    = FA*DALP  + FB*DBET  + FD*DIS
278      IF(ABS(F).LT.TOLF) GO TO 310
279      300 CONTINUE
280      310 CONTINUE
281
282      C          CHECK TO SEE IF MOVEMENT IN THESC, PHISC AND THESC2 IS LESS THAN
283      C          TOLANG IN LAST ITERATION
284      IF ( ABS(THESC-THESSAV).LT.TOLANG
285      * .AND.ABS(PHISC-PHISSAV).LT.TOLANG
286      * .AND.ABS(THESC2-THESS2SV).LT.TOLANG) GO TO 1010
287      THESSAV=THESC
288      PHISSAV=PHISC
289      THESS2SV=THESC2
290      1000 CONTINUE
291      1010 CONTINUE
292
293      IF (IREPT.GE.12) GO TO 3020
294
295      PRINT 97, ICONV
296      PRINT 94
297      PRINT 99, THESC, PHISC, THESC2, PHISC2,ALP, BET,DY,DZ
298
299      C          CHECK TO SEE IF CONVERGED OUTSIDE PHIS LIMIT
300      IF(PHISC.GT.PHISLU) THEN
301      PHISC=PHISC-180.
302      THESC=-THESC

```

```

303         GO TO 50
304     END IF
305     IF(PHISC.LT.PHISLL) THEN
306         PHISC=PHISC+180.
307         THESC=-THESC
308         GO TO 50
309     END IF
310
311     C             SAVE ANSWERS UNLIMITED IN PITCH FOR NEXT COMPUTATION
312         THESCVL=THESC
313         PHISCVL=PHISC
314         THES2VL=THESC2
315
316     C             CHECK TO SEE IF THESC CONVERGED OUTSIDE THESC LIMIT
317
318         IF (THESC.LT.THESLL) THEN
319             THESC=THESLL
320             GO TO 2020
321         END IF
322         IF (THESC.GT.THESLU) THEN
323             THESC=THESLU
324             GO TO 2020
325         END IF
326
327         PRINT 98
328         PRINT 96
329         PRINT 99,THESC,PHISC,THESC2,PHISC2,ALPC,BETC,PSIOFF,THEOFF,
330     *             PHIOFF
331
332     C             FINAL CONVERGENCE ON ALPHA COMMAND (ALPC) AND BETA COMMAND (BETC)
333     C             WITHOUT REGARD TO DISTANCE (DIS)
334
335         DO 2000 ICONV=1,100
336
337
338         THESCSV=THESC-1.
339         CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
340     *             THESCSV,PHISC ,THESC2 ,PHISC2,PSIMT,THEMT,PHIMT)

```

```

341      THEMOBT=THEMT+DTHEMOB
342      PHIMOBT=PHIMT+DPHIMOB
343      CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALPSV,BETSV)
344      DALPSV=ALPSV-ALPC
345      DBETSV=BETSV-BETC
346      CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
347      *          THESC ,PHISC ,THESC2 ,PHISC2,PSIMT,THEMT,PHIMT)
348      THEMOBT=THEMT+DTHEMOB
349      PHIMOBT=PHIMT+DPHIMOB
350      CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALP ,BET )
351      DALP =ALP - ALPC
352      DBET =BET - BETC
353      FA=ALP - ALPSV
354      FB=BET - BETSV
355      FSV = FA*DALPSV + FB*DBETSV
356      F   = FA*DALP  + FB*DBET
357      DO 1100 ICTHE=1,3
358      CALL CONV(THESC ,F,THESCSV,FSV)
359      CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
360      *          THESC ,PHISC ,THESC2 ,PHISC2,PSIMT,THEMT,PHIMT)
361      THEMOBT=THEMT+DTHEMOB
362      PHIMOBT=PHIMT+DPHIMOB
363      CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALP ,BET )
364      DALP =ALP - ALPC
365      DBET =BET - BETC
366      F   = FA*DALP  + FB*DBET
367      IF(ABS(F).LT.TOLF) GO TO 1110
368      1100 CONTINUE
369      1110 CONTINUE
370      IF(THESC.EQ.0.) THESC=.000001
371      PHISCSV=PHISC-1.
372      CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
373      *          THESC ,PHISCSV,THESC2 ,PHISC2,PSIMT,THEMT,PHIMT)
374      THEMOBT=THEMT+DTHEMOB
375      PHIMOBT=PHIMT+DPHIMOB
376      CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALPSV,BETSV)
377      DALPSV=ALPSV-ALPC
378      DBETSV=BETSV-BETC

```

```

379      CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
380      *          THESC ,PHISC ,THESC2 ,PHISC2,PSIMT,THEMT,PHIMT)
381      THEMOBT=THEMT+DTHEMOB
382      PHIMOBT=PHIMT+DPHIMOB
383      CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALP ,BET )
384      DALP=ALP-ALPC
385      DBET=BET-BETC
386      FA=ALP-ALPSV
387      FB=BET-BETSV
388      FSV = FA*DALPSV + FB*DBETSV
389      F   = FA*DALP  + FB*DBET
390      DO 1200 ICPHI=1,3
391      CALL CONV(PHISC,F,PHISCSV,FSV)
392      CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
393      *          THESC ,PHISC ,THESC2 ,PHISC2,PSIMT,THEMT,PHIMT)
394      THEMOBT=THEMT+DTHEMOB
395      PHIMOBT=PHIMT+DPHIMOB
396      CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALP ,BET )
397      DALP=ALP-ALPC
398      DBET=BET-BETC
399      F   = FA*DALP  + FB*DBET
400      IF(ABS(F).LT.TOLF) GO TO 1210
401      1200 CONTINUE
402      1210 CONTINUE
403      THESC2S=THESC2-1.
404      CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
405      *          THESC ,PHISC ,THESC2S,PHISC2,PSIMT,THEMT,PHIMT)
406      THEMOBT=THEMT+DTHEMOB
407      PHIMOBT=PHIMT+DPHIMOB
408      CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALPSV,BETSV)
409      DALPSV=ALPSV-ALPC
410      DBETSV=BETSV-BETC
411      CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
412      *          THESC ,PHISC ,THESC2 ,PHISC2,PSIMT,THEMT,PHIMT)
413      THEMOBT=THEMT+DTHEMOB
414      PHIMOBT=PHIMT+DPHIMOB
415      CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALP ,BET )
416      DALP=ALP-ALPC

```

```

417         DBET=BET-BETC
418         FA=ALP -ALPSV
419         FB=BET -BETSV
420         FSV = FA*DALPSV + FB*DBETSV
421         F   = FA*DALP  + FB*DBET
422         DO 1300 ICTHE2=1,3
423         CALL CONV(THESC2,F,THESC2S,FSV)
424         CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
425         *      THESC ,PHISC ,THESC2 ,PHISC2,PSIMT,THEMT,PHIMT)
426         THEMOBT=THEMT+DTHEMOB
427         PHIMOBT=PHIMT+DPHIMOB
428         CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALP ,BET )
429         DALP=ALP-ALPC
430         DBET=BET-BETC
431         F   = FA*DALP  + FB*DBET
432         IF(ABS(F).LT.TOLF) GO TO 1310
433         1300 CONTINUE
434         1310 CONTINUE
435
436         C           CHECK TO SEE IF ALP AND BET ARE WITHIN TOLDAB OF ALPC AND BETC
437         IF ( (ABS(DALP)+ABS(DBET)) .LT. TOLDAB ) GO TO 2010
438
439         2000 CONTINUE
440         2010 CONTINUE
441
442         IF (IREPT.GE.6) GO TO 3020
443
444         PRINT 97, ICONV
445         PRINT 94
446         PRINT 99, THESC, PHISC, THESC2, PHISC2,ALP, BET
447
448         C           CHECK TO SEE IF THESC CONVERGED OUTSIDE THESC LIMIT
449
450         IF (THESC.LT.THESLL) THEN
451             THESC=THESLL
452             GO TO 2020
453         END IF
454         IF (THESC.GT.THESLU) THEN

```

```

455         THESC=THESLU
456         GO TO 2020
457     END IF
458
459     C             CHECK TO SEE IF CONVERGED OUTSIDE PHIS LIMIT
460     IF(PHISC.GT.PHISLU) THEN
461         PHISC=PHISC - 180.
462         THESC= -THESC
463         GO TO 50
464     END IF
465     IF(PHISC.LT.PHISLL) THEN
466         PHISC=PHISC + 180.
467         THESC= -THESC
468         GO TO 50
469     END IF
470
471     C             GO TO END
472     GO TO 3020
473
474     2020 CONTINUE
475     C             FINAL CONVERGENCE ON ALPHA COMMAND (ALPC) AND BETA COMMAND (BETC)
476     C             WITHOUT REGARD TO DISTANCE (DIS), AND WITH THESC SET TO
477     C             THE LIMIT (THESLL OR THESLU)
478
479     C             SET TO PREVIOUS ANSWERS LIMITED IN THESC
480     PHISC = PHISCL
481     THESC2 = THESC2L
482
483         PRINT 98
484         PRINT 96
485         PRINT 99,THESC,PHISC,THESC2,PHISC2,ALPC,BETC,PSIOFF,THEOFF,
486     *             PHIOFF
487
488     DO 3000 ICONV=1,100
489
490     THESC2S=THESC2-1.
491     CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
492     *             THESC ,PHISC ,THESC2S,PHISC2,PSINT,THEMT,PHIMT)

```

```

493      THEMOBT=THEMT+DTHEMOB
494      PHIMOBT=PHIMT+DPHIMOB
495      CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALPSV,BETSV)
496      DALPSV=ALPSV-ALPC
497      DBETSV=BETSV-BETC
498      CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
499      *          THESC ,PHISC ,THESC2 ,PHISC2,PSIMT,THEMT,PHIMT)
500      THEMOBT=THEMT+DTHEMOB
501      PHIMOBT=PHIMT+DPHIMOB
502      CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALP ,BET )
503      DALP=ALP-ALPC
504      DBET=BET-BETC
505      FA=ALP -ALPSV
506      FB=BET -BETSV
507      FSV = FA*DALPSV + FB*DBETSV
508      F   = FA*DALP  + FB*DBET
509      DO 2300 ICTHE2=1,3
510      CALL CONV(THESC2,F,THESC2S,FSV)
511      CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
512      *          THESC ,PHISC ,THESC2 ,PHISC2,PSIMT,THEMT,PHIMT)
513      THEMOBT=THEMT+DTHEMOB
514      PHIMOBT=PHIMT+DPHIMOB
515      CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALP ,BET )
516      DALP=ALP-ALPC
517      DBET=BET-BETC
518      F   = FA*DALP  + FB*DBET
519      IF(ABS(F).LT.TOLF) GO TO 2310
520      2300 CONTINUE
521      2310 CONTINUE
522
523      IF(THESC.EQ.0.) THESC=.000001
524      PHISCSV=PHISC-1.
525      CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
526      *          THESC ,PHISCSV,THESC2 ,PHISC2,PSIMT,THEMT,PHIMT)
527      THEMOBT=THEMT+DTHEMOB
528      PHIMOBT=PHIMT+DPHIMOB
529      CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALPSV,BETSV)
530      DALPSV=ALPSV-ALPC

```

```

531         DBETSV=BETSV-BETC
532         CALL SIMUST(PSTOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
533         *          THESC ,PHISC ,THESC2 ,PHISC2,PSIMT,THEMT,PHIMT)
534         THEMOBT=THEMT+DTHEMOB
535         PHIMOBT=PHIMT+DPHIMOB
536         CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALP ,BET )
537         DALP=ALP-ALPC
538         DBET=BET-BETC
539         FA=ALP-ALPSV
540         FB=BET-BETSV
541         FSV = FA*DALPSV + FB*DBETSV
542         F   = FA*DALP  + FB*DBET
543         DO 2200 ICPHI=1,3
544         CALL CONV(PHISC,F,PHISCSV,FSV)
545         CALL SIMUST(PSTOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
546         *          THESC ,PHISC ,THESC2 ,PHISC2,PSIMT,THEMT,PHIMT)
547         THEMOBT=THEMT+DTHEMOB
548         PHIMOBT=PHIMT+DPHIMOB
549         CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALP ,BET )
550         DALP=ALP-ALPC
551         DBET=BET-BETC
552         F   = FA*DALP  + FB*DBET
553         IF(ABS(F).LT.TOLF) GO TO 2210
554         2200 CONTINUE
555         2210 CONTINUE
556         C          CHECK TO SEE IF ALP AND BET ARE WITHIN TOLDAB OF ALPC AND BETC
557         IF ( (ABS(DALP)+ABS(DBET)) .LT. TOLDAB ) GO TO 3010
558
559         3000 CONTINUE
560         3010 CONTINUE
561
562         IF (IREPT.GE.6) GO TO 3020
563
564         PRINT 97, ICONV
565         PRINT 94
566         PRINT 99, THESC, PHISC, THESC2, PHISC2,ALP, BET
567
568         C          CHECK TO SEE IF CONVERGED OUTSIDE THIS LIMIT AND SAVE ANSWERS

```

```

569           C           WITH PITCH LIMITED TO THESCLL OR THESCLU FOR NEXT COMPUTATION
570           IF(PHISC.GT.PHISLU) THEN
571             PHISC=PHISC - 180.
572             THESC= -THESC
573             PHISCL=PHISC
574             THESC2L=THESC2
575             GO TO 50
576           END IF
577           IF(PHISC.LT.PHISLL) THEN
578             PHISC=PHISC + 180.
579             THESC= -THESC
580             PHISCL=PHISC
581             THESC2L=THESC2
582             GO TO 50
583           END IF
584
585           C           SAVE ANSWERS WITH PITCH LIMITED TO THESLL OR THESLU
586           C           FOR NEXT COMPUTATION
587             PHISCL=PHISC
588             THESC2L=THESC2
589           3020 CONTINUE
590             CALL DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESC,PHISC,THESC2 ,
591             *           DY ,DZ )
592             PRINT 98
593             PRINT 94
594             PRINT 99, THESC, PHISC, THESC2, PHISC2,ALP, BET,DY,DZ
595             PRINT 98
596             PRINT 98
597             STOP
598           99   FORMAT (6F9.4,2F7.3,F10.5)
599           98   FORMAT ( )
600           97   FORMAT(1H ,"ICONV =",I3)
601           96   FORMAT("  THESC   PHISC   THESC2   PHISC2   ALPC   BETC",
602             *     "  PSIOFF  THEOFF  PHIOFF")
603           95   FORMAT("  THESC   PHISC  THESC2   PHISC2   DALP",
604             *     "  DBET    DY     DZ     F")
605           94   FORMAT("  THESC   PHISC  THESC2   PHISC2   ALP   BET ",
606             *     "  DY     DZ")

```

```
607          93  FORMAT (6F9.4,14X,F10.5)
608          92  FORMAT("  THEMOb  PHIMOb  DTHEMOB  DPHIMOb")
609          END
```

```

1          SUBROUTINE SIMUST (PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
2          *                   THES,PHIS,THES2,PHIS2,PSIMT,THEMT,PHIMT)
3
4          C THIS SUBROUTINE SIMULATES THE STING. IT CALCULATES THE THEORETICAL MODEL
5          C YAW, PITCH AND ROLL GIVEN INPUTS OF THE STING OFFSETS, STING BENDING,
6          C STING ROLL AND PITCH, AND SECOND STING ROLL AND PITCH
7          DOR=57.2957795
8
9          CALL COMP (1.,0.,0.,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
10         *          THES,PHIS,THES2,PHIS2,XX,YX,ZX)
11         CALL COMP (0.,1.,0.,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
12         *          THES,PHIS,THES2,PHIS2,XY,YY,ZY)
13         CALL COMP (0.,0.,1.,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
14         *          THES,PHIS,THES2,PHIS2,XZ,YZ,ZZ)
15
16
17         C          CALCULATE PITCH OF MODEL
18         IF(XZ.LT.-1.)XZ=-1.
19         IF(XZ.GT. 1.)XZ= 1.
20         THEMT=ASIN(-XZ)*DOR
21
22         C          ROLL AND YAW OF THE MODEL
23         IF(YZ.EQ.0..AND.ZZ.EQ.0.) THEN
24         C          CASE WHERE PITCH OF MODEL IS +/-90.
25         PHIMT=ATAN2(-YX,ZX)*DOR
26         PSIMT=0.
27         ELSE
28         PHIMT=ATAN2(-YZ,ZZ)*DOR
29         PSIMT=ATAN2(-XY,XX)*DOR
30         END IF
31         RETURN
32         END

```

```

1          SUBROUTINE COMP (X,Y,Z,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
2          *                THES,PHIS,THES2,PHIS2,XK,YK,ZK)
3
4          C
5          C THIS SUBROUTINE IS USED TO CALCULATE THE COMPONENTS, IN THE MODEL
6          C AXIS SYSTEM, OF A VECTOR (X,Y,Z) IN THE WIND TUNNEL AXIS SYSTEM
7          C
8          C          STING PITCH (Y)
9          XB=X*COSD(THES)-Z*SIND(THES)
10         YB=Y
11         ZB=Z*COSD(THES)+X*SIND(THES)
12         C          STING ROLL (X)
13         XC=XB
14         YC=YB*COSD(PHIS)-ZB*SIND(PHIS)
15         ZC=ZB*COSD(PHIS)+YB*SIND(PHIS)
16         C          OFFSET YAW (Z)
17         XD=XC*COSD(PSIOFF)-YC*SIND(PSIOFF)
18         YD=YC*COSD(PSIOFF)+XC*SIND(PSIOFF)
19         ZD=ZC
20         C          OFFSET PITCH (Y)
21         XE=XD*COSD(THEOFF)-ZD*SIND(THEOFF)
22         YE=YD
23         ZE=ZD*COSD(THEOFF)+XD*SIND(THEOFF)
24         C          OFFSET ROLL (X)
25         XF=XE
26         YF=YE*COSD(PHIOFF)-ZE*SIND(PHIOFF)
27         ZF=ZE*COSD(PHIOFF)+YE*SIND(PHIOFF)
28         C          SECOND STING PITCH (Y)
29         XG=XF*COSD(THES2)-ZF*SIND(THES2)
30         YG=YF
31         ZG=ZF*COSD(THES2)+XF*SIND(THES2)
32         C          SECOND STING ROLL (X)
33         XH=XG
34         YH=YG*COSD(PHIS2)-ZG*SIND(PHIS2)
35         ZH=ZG*COSD(PHIS2)+YG*SIND(PHIS2)
36         C          STING BENDING IN YAW (Z)
          XI=XH*COSD(PSISB)-YH*SIND(PSISB)

```

```

37      YI=YH*COSD(PSISB)+XH*SIND(PSISB)
38      ZI=ZH
39      C      STING BENDING IN PITCH (Y)
40      XJ=XI*COSD(THESB)-ZI*SIND(THESB)
41      YJ=YI
42      ZJ=ZI*COSD(THESB)+XI*SIND(THESB)
43      C      STING BENDING IN ROLL (X)
44      XK=XJ
45      YK=YJ*COSD(PHISB)-ZJ*SIND(PHISB)
46      ZK=ZJ*COSD(PHISB)+YJ*SIND(PHISB)
47      RETURN
48      END

```

```

1      SUBROUTINE ALPBET(U,V,W,PSIM,THEM,PHIM,ALP,BET)
2      C      THIS SUBROUTINE CALCULATES THE ANGLE OF ATTACK (ALP) AND
3      C      ANGLE OF SIDESLIP (BET) OF A WIND TUNNEL MODEL.  THE INPUTS ARE
4      C      THE VELOCITY COMPONENTS IN THE WIND TUNNEL (U, V, AND W), AND THE
5      C      MODEL EULER ROTATION ANGLES, YAW (PSIM), PITCH (THEM) AND ROLL (PHIM).
6      C      THE SUBROUTINE CALCULATES THE COMPONENTS OF THE FREE STREAM VELOCITY
7      C      ALONG THE THREE AXES OF THE MODEL AFTER EACH ROTATION AND THESE VELOCITY
8      C      COMPONENTS ARE USED TO CALCULATE THE ANGLE OF ATTACK AND ANGLE OF SIDESLIP.
9      C
10     DOR=57.2957795
11     C      YAW (Z)
12     UA=U *COSD(PSIM)-V *SIND(PSIM)
13     VA=V *COSD(PSIM)+U *SIND(PSIM)
14     WA=W
15     C      PITCH (Y)
16     UB=UA*COSD(THEM)-WA*SIND(THEM)
17     VB=VA
18     WB=WA*COSD(THEM)+UA*SIND(THEM)
19     C      ROLL (X)
20     UC=UB
21     VC=VB*COSD(PHIM)-WB*SIND(PHIM)
22     WC=WB*COSD(PHIM)+VB*SIND(PHIM)
23     C      ALPHA AND BETA
24     IF(WC.NE.0..AND.UC.NE.0.) THEN
25         ALP=ATAN2(WC,UC)*DOR
26     ELSE
27         ALP=0.
28     END IF
29     IF(VC.LT.-1.)VC=-1.
30     IF(VC.GT. 1.)VC= 1.
31     BET=ASIN(-VC)*DOR
32     RETURN
33     END

```

```

1          SUBROUTINE DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESC,PHISC,THESC2,
2          *          DY,DZ)
3
4          C          THIS SUBROUTINE CALCULATES THE POSITION OF THE MODEL CENTER
5          C WITH RESPECT TO THE TUNNEL AXIS SYSTEM.  THE SUBROUTINE STARTS AT
6          C THE MODEL AND PROCEEDS TO THE TUNNEL AXIS SYSTEM.  THE ROTATIONS
7          C AND TRANSLATIONS ARE TAKEN IN REVERSE ORDER FROM THE ORDER USED
8          C IN THE MAIN PROGRAM.  THE STING BENDING ANGLES ARE IGNORED.
9          C THE ORDER OF ROTATIONS ARE:
10         C SECOND STING ROLL (PHISC2), SECOND STING PITCH (THESC2),
11         C OFFSET ROLL (PSIOFF), OFFSET PITCH (THEOFF), OFFSET YAW (PSIOFF),
12         C STING ROLL (PHISC), AND STING PITCH (THESC).
13         C          R1 IS THE RADIUS OF ROTATION OF THE ARC SECTOR.
14         C          X2 IS THE LENGTH OF THE MAIN STING.
15         C          X3 IS THE LENGTH OF THE SECOND (ARTICULATED) STING.
16         C
17         C          XA=0.
18         C          YA=0.
19         C          ZA=0.
20
21         C          TRANSLATION (X) HIGH ALPHA STING
22         C          XB=XA - X3
23         C          YB=YA
24         C          ZB=ZA
25
26         C          2ND STING ROLL (X)
27         C          NO EFFECT ON X, Y OR Z
28
29         C          2ND STING PITCH (Y)
30         C          XC=XB*COSD(-THESC2)-ZB*SIND(-THESC2)
31         C          YC=YB
32         C          ZC=ZB*COSD(-THESC2)+XB*SIND(-THESC2)
33
34         C          TRANSLATION (X) MAIN STING
35         C          XD=XC - X2
36         C          YD=YC

```

```

37          ZD=ZC
38
39          C          OFFSET ROLL (X)
40          XE=XD
41          YE=YD*COSD(-PHIOFF)-ZD*SIND(-PHIOFF)
42          ZE=ZD*COSD(-PHIOFF)+YD*SIND(-PHIOFF)
43
44          C          OFFSET PITCH (Y)
45          XF=XE*COSD(-THEOFF)-ZE*SIND(-THEOFF)
46          YF=YE
47          ZF=ZE*COSD(-THEOFF)+XE*SIND(-THEOFF)
48
49          C          OFFSET YAW (Z)
50          XG=XF*COSD(-PSIOFF)-YF*SIND(-PSIOFF)
51          YG=YF*COSD(-PSIOFF)+XF*SIND(-PSIOFF)
52          ZG=ZF
53
54          C          TRANSLATION (X) ,ARC SECTOR
55          XH=XG + R1
56          YH=YG
57          ZH=ZG
58
59          C          STING ROLL (X)
60          XI=XH
61          YI=YH*COSD(-PHISC)-ZH*SIND(-PHISC)
62          ZI=ZH*COSD(-PHISC)+YH*SIND(-PHISC)
63
64          C          STING PITCH (Y)
65          XJ=XI*COSD(-THESC)-ZI*SIND(-THESC)
66          YJ=YI
67          ZJ=ZI*COSD(-THESC)+XI*SIND(-THESC)
68
69          DY=YJ
70          DZ=ZJ
71
72          RETURN
73          END

```

```

1          SUBROUTINE CONV (X,Y,XSAVE,YSAVE)
2      C
3      C THIS SUBROUTINE CALCULATES THE VALUE OF X WHICH MINIMIZES THE VALUE OF Y.
4      C THE CALLING PROGRAM CALCULATES THE VALUES OF Y FOR EACH VALUE OF X.
5      C THE NEW VALUE OF X IS DETERMINED BY CALCULATING WHERE A STRAIGHT LINE
6      C THROUGH THE LAST TWO PREVIOUS PAIR OF POINTS, X AND Y, AND XSAVE AND YSAVE,
7      C INTERSECT THE X AXIS. THIS NEW VALUE OF X IS RETURNED TO THE CALLING
8      C PROGRAM IN THE X PARAMETER.
9      C
10         X1=XSAVE
11         Y1=YSAVE
12         X2=X
13         Y2=Y
14         IF(X2.EQ.X1) GO TO 3
15         IF(Y2.EQ.Y1) GO TO 4
16         XK=(Y2-Y1)/(X2-X1)
17         X=X2-Y2/XK
18         XA=(X1+X2)/2.
19         DX=ABS(X2-X1)
20         IF(ABS(X-XA).GT.3.5*DX) X=XA+3.5*SIGN( DX, (X-XA) )
21         GO TO 100
22     3     X=X2-Y2
23         GO TO 100
24     4     X=(X2+X1)/2.
25     100   XSAVE=X2
26         YSAVE=Y2
27         IF(ABS(Y1).LT.ABS(Y2)) THEN
28             XSAVE=X1
29             YSAVE=Y1
30         END IF
31         RETURN
32     END

```

PSI = ψ = Euler yaw angle = $\angle ABC$
 THETA = θ = Euler pitch angle = $\angle C&D$ (Note $\theta \neq \alpha$ unless $\phi = 0^\circ$)
 PHI = ϕ = Euler roll angle = $\angle CDE$

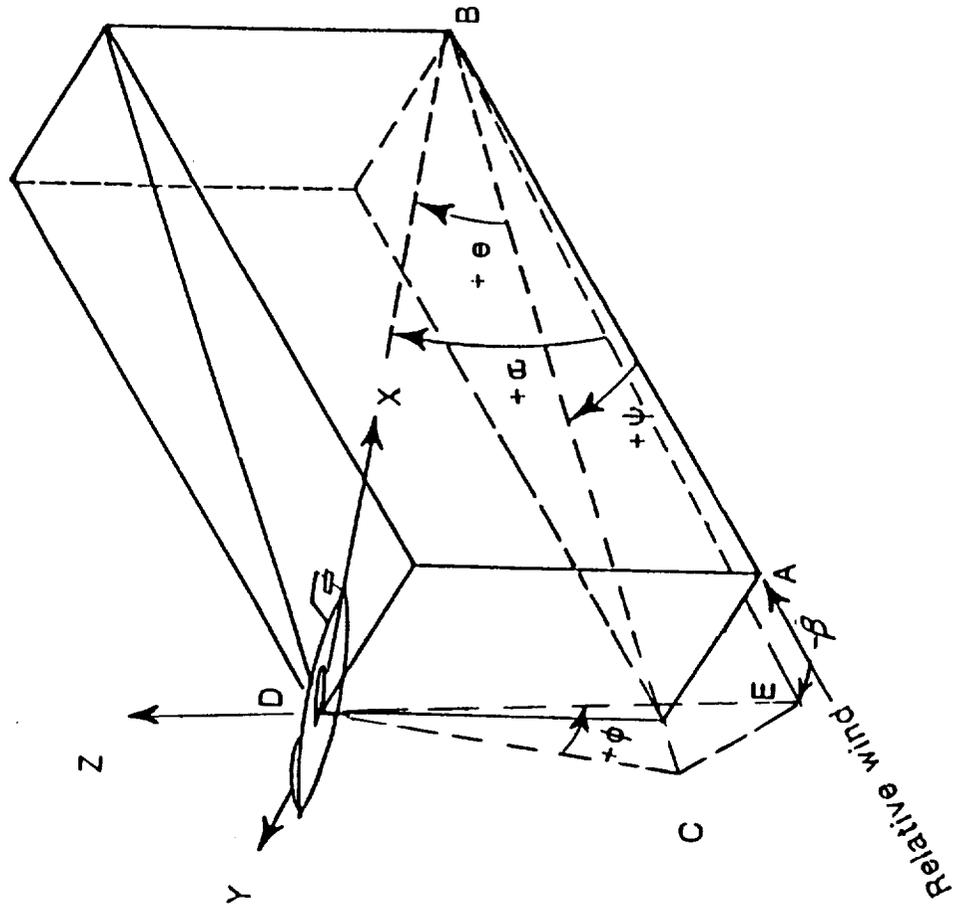


Figure 1. Definition of Euler angles and directions

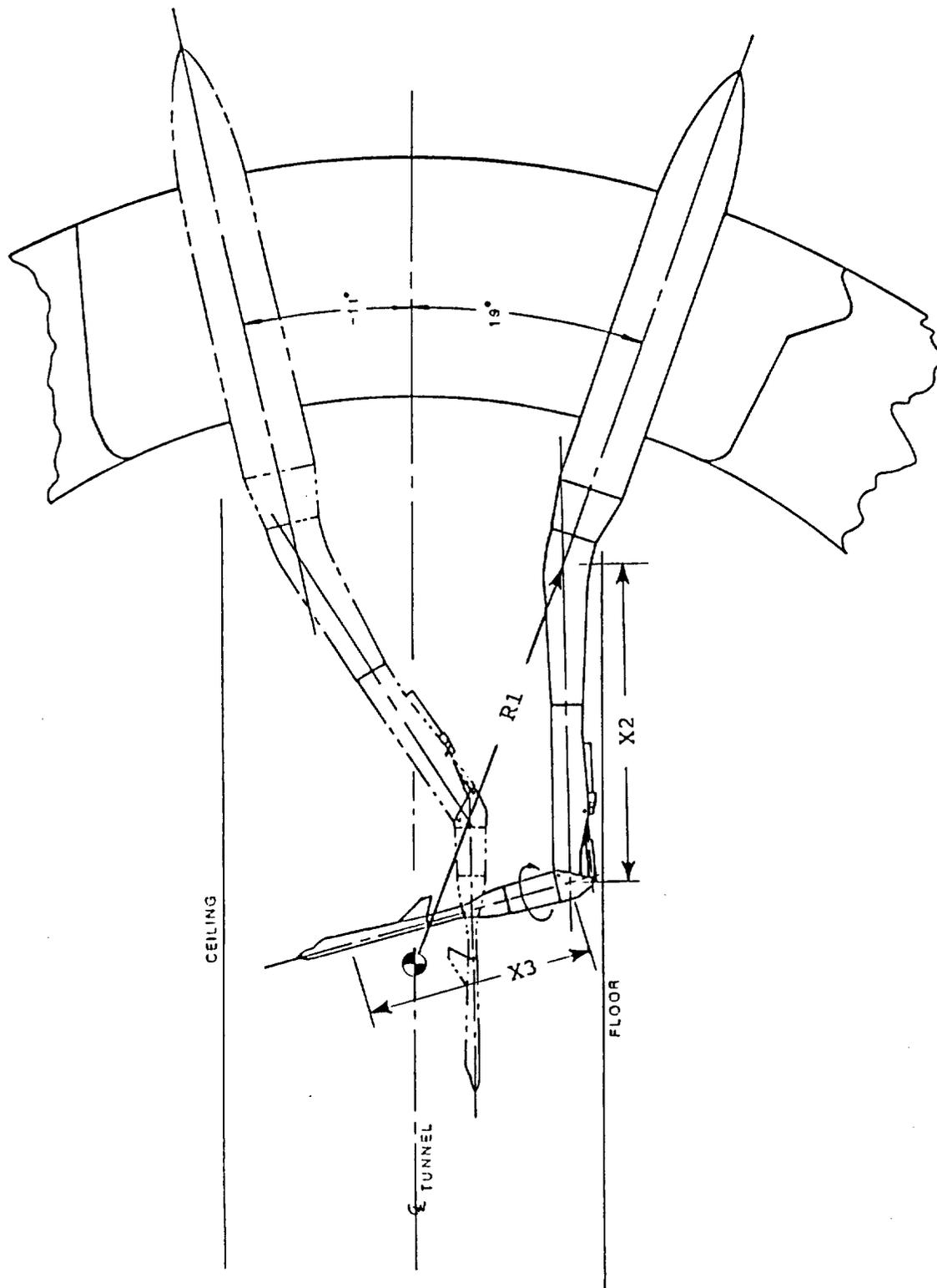


Figure 2. - High angle of attack articulated sting mechanism

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE October 1991	3. REPORT TYPE AND DATES COVERED Technical Memorandum	
4. TITLE AND SUBTITLE Computer Programs for the Calculation of Dual Sting Pitch and Roll Angles Required for an Articulated Sting to Obtain Angles of Attack and Sideslip on Wind-Tunnel Models			5. FUNDING NUMBERS WU 505-59-54-01	
6. AUTHOR(S) John B. Peterson, Jr.				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) NASA Langley Research Center Hampton, VA 23665-5225			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, DC 20546-0001			10. SPONSORING/MONITORING AGENCY REPORT NUMBER NASA TM-104161	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Unclassified-unlimited Subject category-09			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) Two programs have been developed to calculate the pitch and roll position of the conventional sting drive and the pitch of a high angle articulated sting to position a wind tunnel model at the desired angle of attack and sideslip and position the model as near as possible to the centerline of the tunnel. These programs account for the effects of sting offset angles, sting bending angles, and wind-tunnel stream flow angles. In addition, the second program incorporates inputs from on-board accelerometers that measure model pitch and roll with respect to gravity. The programs are presented in the report and a description of the numerical operation of the programs with a definition of the variables used in the programs is given.				
14. SUBJECT TERMS Wind-tunnel apparatus; Support systems; Computer programs; Angle of attack; Roll; Sideslip; Attitude control; Pitch; Mounts; Yaw; Rotation; and Supports; Struts			15. NUMBER OF PAGES 74	
			16. PRICE CODE A04	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT Unlimited	



